FACULTY OF ENGINEERING
AND ARCHITECTURE

Advanced Software-Defined Radio Architectures and Techniques for Radio Hardware Virtualization and Efficient Spectrum Sharing

Felipe Augusto Pereira de Figueiredo

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Electrical Engineering

Supervisors
Prof. Ingrid Moerman, PhD - Xianjun Jiao, PhD

Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

September 2021

GHENT
UNIVERSITY

FACULTY OF ENGINEERING
AND ARCHITECTURE

Advanced Software-Defined Radio Architectures and Techniques for Radio Hardware Virtualization and Efficient Spectrum Sharing

Felipe Augusto Pereira de Figueiredo

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Electrical Engineering

Supervisors
Prof. Ingrid Moerman, PhD - Xianjun Jiao, PhD

Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

September 2021

GHENT
UNIVERSITY

ISBN 978-94-6355-517-3
NUR 959
Wettelijk depot: D/2021/10.500/65

Members of the Examination Board

Chair

Prof. Em. Daniël De Zutter, PhD, Ghent University

Other members entitled to vote

Spilios Giannoulis, PhD, Ghent University
Prof. Mamoun Guenach, PhD, Ghent University
Prof. Sofie Pollin, PhD, KU Leuven
Justin Tallon, PhD, Software Radio Systems Limited, Ireland
Prof. Guy Torfs, PhD, Ghent University

Supervisors

Prof. Ingrid Moerman, PhD, Ghent University
Xianjun Jiao, PhD, Ghent University

Acknowledgments

First and foremost, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without His blessings, this achievement would not have been possible.

I would like to extend my sincere thanks to my promoters, Prof. Dr. Ingrid Moerman and Dr. Xianjun Jiao for their valuable support and supervision. It has been a great privilege and a pleasure working with them. Their advice, guidance, and encouragement have been invaluable during the course of my research.

I would also like to give special thanks to Dr. Jetmir Haxhibeqiri, Dr. Vasilis Maglogiannis, and Dr. Abdulkadir Karaağaç for their true friendship and for sharing their invaluable expertise on several aspects of my work.

Nevertheless, none of what I have achieved in my life would have been possible without the tremendous support of my mother, Aldair Ibrantina Pereira de Figueiredo, who has always encouraged me to keep going no matter what happened.

Additionally, in a unique way, and most importantly, I would like to thank my wife, Samanta Elena Moreira de Figueiredo, for all her dedication to contributing to my life. Her support, encouragement, patience and unconditional love were undeniably the foundation on which the past seven years of my life have been built upon. She has also given me, two years ago, the most precious gift anyone could ever receive, my daughter, Julia Elena Moreira de Figueiredo, which is a true jewel.

Gent, August 2021
Felipe Augusto Pereira de Figueiredo

Table of Contents

List of Figures

List of Tables

List of Acronyms

0-9

2G	2nd Generation
3G	3rd Generation
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
4G	4th Generation
5G	5th Generation
5G-NR	5G New-Radio

A

ACK	Acknowledgment
API	Application Programming Interface
AP	Access Point
ASIP	Application-Specific Instruction set Processor
ASIC	Application-Specific Integrated Circuit
ADC	Analog to Digital Converter
AXI	Advanced Extensible Interface
AWGN	Additive white Gaussian Noise
AI	Artificial Intelligence
ACLR	Adjacent Channel Leakage Ratio
AVX	Advanced Vector Extension

B

BBU	Baseband Unit
BLE	Bluetooth Low Energy
BW	Bandwidth
BER	Bit Error Rate
BDMA	beam division multiple access
BS	base station
BLER	block error rate

C

CA	Cell Averaging
C-RAN	Cloud/Centralized Radio Access Networks
CPU	Central Programming Unit
CPRI	Common Public Radio Interface
CAPEX	Capital Expenditure
CQI	Channel Quality Indicator
CRC	cyclic redundancy check
CAGR	Compound Annual Growth Rate
CRAN	cloud-based radio access networks
CP	cyclic prefix
CBRS	Citizens Broadband Radio Service
CIC	cascaded integrator-comb
C-V2X	Cellular Vehicle-to-Anything
COT	Channel Occupancy Time
CFO	Carrier Frequency Offset
CA-CFAR	Cell-Average Constant False Alarm Rate
CCA	Clear Channel Assessment
CSAT	Carrier-Sensing Adaptive Transmission
CW	Contention Window
COTS	commercially available off-the-shelf
CRS	Cell-Specific Reference Signal
CSMA/CA	Carrier-sense multiple access with collision avoidance

CDF	cumulative distribution function

D

DAC	Digital to Analog Converter
DDC	Digital Down-converter
DMA	Direct Memory Access
DSA	Dynamic Spectrum Access
DC	Direct Current
DUT	device under test
DVB-T	Digital Video Broadcasting-Terrestrial
DLMA	Deep-reinforcement Learning Multiple Access
DARPA	Defense Advanced Research Projects Agency
DLNN	Deep Learning Neural Network
DAE	Deep Autoencoder
Demod	Demodulation
DSP	Digital Signal Processor
DSS	Dynamic Spectrum Sharing

E

ETSI	European Telecommunications Standards Institute
eMBB	Enhanced Mobile Broadband
ED	energy detection

F

FM	Frequency Modulation
FCME	Forward Consecutive Mean Excision
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FIFO	First-In First-Out
FPGA	Field-Programmable Gate Array

FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FCC	Federal Communications Commission
FSS	Fixed Satellite Services (FSS)
f-OFDM	Filtered-OFDM
FSM	Finite State Machine
FEC	Forward Error Correction

G

GSM	Global System for Mobile Communications
GPU	Graphics Processing Units
Gbps	Gigabits per second
GW	Gateway
GUI	Graphical User Interface
GB	guard-band

H

HARQ	Hybrid Automatic Repeat Request
HLS	High-Level Synthesis
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HW	Hardware
HTC	Human-type communication

I

ID	Identity
IEEE	Institute of Electrical and Electronics Engineers
ILA	Integrated Logic Analyzer
IoT	Internet of Things
ISM	Industrial, Scientific, and Medical

IQ	In-Phase and Quadrature
IF	intermediate-frequency
IFFT	Inverse Fast Fourier Transform
ISI	inter-symbol interference
ICI	inter-carrier interference

K

kbps	kilobit per second

L

LTE	Long-Term Evolution
LTE-A	LTE-Advanced
LTE-LAA	LTE-Licensed Assisted Access
LAA	Licensed Assisted Access
LTE-U	LTE-Unlicensed
LBT	Listen Before Talking
LSA	Licensed Shared Access

M

M-sequence	Maximum-length Sequence
M2M	Machine to Machine
MAC	Medium Access Control
MEC	Mobile Edge Cloud
Mbps	Mega bits per second
Msps	Mega samples per second
MSE	Mean Squared Error
MER	Modulation Error Ratio
MVNO	Mobile Virtual Network Operator
MCS	Modulation Code Scheme
M2M	Machine to Machine

MTC	Machine-type communication
MIMO	multiple-input and multiple-output
mMIMO	massive multiple-input-multiple-output
MF-TDMA	Multiple-Frequency Time Division Multiple Access
McF-TDMA	Multi-Concurrent-Frequency Time-Division Multiple Access
mMTC	Massive Machine Type Communications
ML	machine learning
MAPE	Mean Absolute Percentage Error
M-sequence	Maximum length sequence

N

NIST	Institute of standards and technology
NFV	Network Function Virtualization
NB-IoT	Narrow Band Internet of Things
NR	New Radio
NC-OFDM	Non-Contiguous OFDM
NAS	Non-Access Stratum
NN	Neural Network
NACK	Negative Acknowledgment
NI	National Instruments

O

OAI	Open Air Interface
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OSI	Open Systems Interconnection
OPEX	OPerational Expenditure
OAI	Open Air Interface
OOB	Out-Of-Band
OOBE	Out-Of-Band Emissions

P

PL	Programmable Logic
PS	Processor System
PHY	Physical
PDCCH	Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PCFICH	Physical Control Format Indicator Channel
PHICH	Physical Hybrid ARQ Indicator Channel
PUSCH	Physical Uplink Shared Channel
PRR	Packet Reception Ratio
P2P	peer-to-peer
PDCP	Packet Data Convergence Protocol
PA	power amplifier
PSS	Synchronization Signal
PDU	Payload Data Unit
PSD	Power Spectral Density
PRB	physical resource block
PDF	probability distribution function
PSR	Peak-to-Side-lobe Ratio
PDP	Power Delay Profile

Q

QoS	Quality of Service
QAM	Quadrature Amplitude Modulation
QPP	Quadratic Permutation Polynomial

R

RAN	Radio Access Network
RAT	Radio Access Technologies

RF	Radio Frequency
RRH	Remote Radio Head
RLC	Radio Link Control
RRC	Radio Resource Control
RRM	Radio Resource Management
RANaaS	RAN as a service
RSSI	Received Signal Strength Indication
RB	resource blocks
RSRP	Reference Signal Received Power
RE	resource element
RL	Reinforcement Learning
RTT	round trip time

S

SDN	Software-Defined Network
SDWN	Software-Defined Wireless Network
SDR	Software-Defined Radio
SIFS	Short Inter Frame Space
SoC	System-on-Chip
SW	Software
S1AP	S1 Application Protocol
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
SIMD	Single Instruction Multiple Data
SSS	Secondary Synchronization Signal
SRS	Software Radio Systems
SINR	signal-to-interference-plus-noise ratio
SPTF	Spectrum Policy Task Force
SAS	Spectrum Access Systems
SC2	Spectrum Collaboration Challenge
SL	Supervised Learning
SSL	Semi-Supervised Learning
Synch	Synchronization

T

TDD	Time Division Duplex
TDM	Time-Division Multiplexing
TDMA	Time Division Multiple Access
TC	Traffic Classes
TTI	Transmission Time Interval
TB	Transport Block
TV	Television

U

UAV	Unmanned Aerial Vehicle
UTP	Unshielded Twisted Pair
USRP	Universal Software Radio Peripheral
UHD	USRP Hardware Driver
URLLC	Ultra Reliable Low Latency Communications

V

VLAN	Virtual Local Area Network
VoIP	Voice over IP
VPN	Virtual Private Network
VR	Virtual Reality
vPHY	virtual PHY
vPHYsaaS	vPHYs as a Service
VSAT	very small aperture terminal
V2x	Vehicle to Everything

W

WAN	Wide Area Network

WARC	World Administrative Radio Conference
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WARP	Wireless Open-Access Research Platform
WNV	wireless networks virtualization

Z

ZC	Zadoff-Chu

Samenvatting – Summary in Dutch –

De beschikbaarheid van alomtegenwoordige draadloze en mobiele netwerken, de opkomst van nieuwe technologieën en standaarden, die verschillende frequentiebereiken dekken, zowel met als zonder vergunning, hebben geleid tot de opkomst van een groot aantal draadloze en mobiele apparaten. Bovendien wordt verwacht dat het aantal aangesloten apparaten binnenkort het aantal mensen ruimschoots zal overtreffen. Door deze uitzonderlijke groei wordt het radiospectrum steeds meer verzadigd, wat leidt tot suboptimaal spectrumgebruik. Veel van deze apparaten bestaan naast elkaar in dezelfde omgeving en helaas is het radiospectrum een schaarse hulpbron waarbij de beschikbare bandbreedte niet meeschaalt met de gevraagde draadloze bandbreedte. Sommige delen van het draadloze radiospectrum zijn overvol (bijv. Licentievrije Industrial, Scientific, and Medical (ISM)-banden), terwijl andere mogelijk aanzienlijk onderbenut zijn (Television (TV)- en Frequency Modulation (FM)-gelicentieerde banden). Daarom is het dringend nodig om met nieuwe architecturen en technieken te komen die kunnen worden gebruikt om op een intelligentere en eerlijkere manier toegang te krijgen tot het radiospectrum en deze te delen. Deze architecturen en technieken moeten gericht zijn op het verbeteren van de coördinatie binnen een bepaalde spectrale band of het verkennen van de mogelijkheden van opportunistisch spectrumgebruik van ongebruikt spectrum in onderbenutte frequentiebanden. Om een betere coördinatie en eerlijker spectrumdeling aan te pakken, moeten daarom geavanceerde SDR-architecturen en -technieken worden ingevoerd voor radio-hardwarevirtualisatie en voor toegang tot en delen van het radiospectrum op een zeer efficiënte, eerlijke en effectieve manier.

Het eerste deel van dit proefschrift, dat is samengesteld in hoofdstuk 1, situeert het uitgevoerde onderzoekswerk, vat de belangrijkste uitdagingen en bijdragen samen en schetst de structuur van dit proefschrift. Het geeft ook een overzicht van de publicaties die tijdens deze onderzoeksperiode zijn geschreven. In dit hoofdstuk geven we de lezer in eerste instantie enige context over de onderwerpen die in dit werk aan bod komen. Ten eerste presenteren we **SWDN!** (**SWDN!**), dat kan worden gezien als de toepassing van SDN-concepten in draadloze en mobiele netwerken, en resource slicing, een techniek die afzonderlijke en onafhankelijke verkeersstromen creëert in afzonderlijke subruimten. In de reeks hebben we het ook over Cloud/Centralized Radio Access Networks (C-RAN) en functioneel, concepten die worden gebruikt om dynamische gedeelde resourcetoewijzing in RAN's te bereiken. We bespreken ook de implementatiebeperkingen die betrokken zijn bij het inzetten van C-RAN's. Vervolgens geven we wat context over het tweede onderwerp dat door dit werk wordt behandeld, namelijk het delen van spectrum. Voor dit onderzoeksthema hebben we het over de schaarste en onderbenutting van het radiospectrum waarmee we vandaag worden geconfronteerd, die wordt veroorzaakt door een strikte regulering van het spectrum en het enorme aantal mobiele en draadloze apparaten die vechten voor een mogelijkheid om te zenden. Vervolgens hebben we het over cognitieve en intelligente radio's, die naar verwachting technologie zullen zijn die het schaarse en onderbenutte radiospectrum zal optimaliseren. Ten slotte introduceren we de Spectrum Collaboration Challenge van Defense Advanced Research Projects Agency (DARPA), die erop gericht was de deelnemers nieuwe strategieën en paradigma's voor spectrumtoegang te laten bedenken waarin radionetwerken autonoom samenwerken en redeneren over hoe ze het radiospectrum dynamisch kunnen delen. In de volgorde van dit hoofdstuk introduceren we, op basis van wat eerder is besproken, de algemene uitdagingen die in dit onderzoek worden aangepakt. De eerste twee uitdagingen zijn RAN-virtualisatie en spectrumvirtualisatie, waarin de moeilijkheden worden besproken die gepaard gaan met het virtualiseren en softwariseren van netwerkbronnen, inclusief fysieke RF-frontends. Vervolgens laten we zien dat de virtualisatie en softwarisering van RF-frontends de uitdagingen van Dynamic Spectrum Sharing (DSS) kunnen aanpakken. De resterende uitdagingen hebben betrekking op het onderzoeksthema voor spectrumdeling, dat ook aan bod kwam in dit doctoraatswerk. We beginnen met het bespreken van de uitdagingen die komen kijken bij een eerlijker spectrum delen en eindigen met het bespreken van de mogelijke uitdagingen die kunnen worden aangepakt door wat we intelligent spectrum delen noemen. Bij het intelligent delen van spectrum worden op Artificial Intelligence (AI) gebaseerde radio's en netwerken gebruikt om het spectrumgebruik te maximaliseren. Om het hoofdstuk af te sluiten, beschrijven we hoe elk overblijvend hoofdstuk van dit proefschrift is samengesteld uit publicaties die zijn uitgevoerd in het kader van dit doctoraat, geven we een lijst van de onderzoeksbijdragen die werden bereikt en beschikbaar gemaakt door die publicaties, en de lijst met onderzoeksresultaten die tijdens dit onderzoek zijn verkregen.

Huidige **SWDN!**-implementaties gebruiken voornamelijk aangepaste firmware, die draait op commerciële embedded applicatieprocessorchips, en zijn beperkt tot lagen boven een lagere Medium Access Control (MAC). Deze beperking komt voort uit het feit dat radio HW meestal specifieke stuurprogramma's vereist, die eigen implementaties hebben. Het is dus vrij complex, zo niet onmogelijk, om in die gevallen virtualisatie en softwarisering van de radio HW te realiseren. Virtualisatie en softwarisering zijn sleuteltechnologieën om tegemoet te komen aan de groeiende vraag naar 5G-datadiensten. Daarom onderzoeken we in hoofdstuk 2 de noodzaak en haalbaarheid van uitbreiding van de virtualisatie en softwarisering van draadloze netwerken naar de radio HW. We stellen een gevirtualiseerde en softwarematige radio HW-zendontvangerarchitectuur voor, die softwarisering en virtualisatietechnieken terugbrengt naar de PHY- en MAC-lagen. Daarnaast bespreken we recente vorderingen in flexibele en generieke PHYlaagimplementaties, de trend naar dynamische spectrumtoewijzingsschema's en real-life **SWDN!**-praktijken. Daarnaast presenteren we ORCA's projectvisie en doelstellingen. Ten slotte beoordelen we latentievereisten en presenteren we een proof-of-concept-demonstratie van de voorgestelde architectuur.

De volgende generatie draadloze en mobiele netwerken zal met verschillende uitdagingen worden geconfronteerd, variërend van een enorme toename van het verkeer tot een efficiënt gebruik van netwerk- en spectrumbronnen. Dit scenario vraagt om nieuwe manieren om de beide bronnen (d.w.z. netwerk en spectrum) te maximaliseren. Daarom stellen we in Hoofdstuk 3 een op SW gebaseerde hypervisor-architectuur voor die een RF-front-end efficiënt abstraheert in een configureerbaar aantal virtuele RF-front-ends. Het maakt flexibele en dynamische spectrumtoegang en delen in bestaande draadloze en mobiele netwerken mogelijk. Dit is een sterk geoptimaliseerde en niet-intrusieve hypervisor-architectuur die de signalen van verschillende en gelijktijdige multi-carrier-gebaseerde RAT's multiplext met gecorreleerde numerologieën. Door gebruik te maken van de voorgestelde architectuur is het mogelijk om de kosten te verlagen en de spectrale efficiëntie te verhogen door verdichting. Om dit hoofdstuk compleet te maken, geven we een wiskundige demonstratie hoe de voorgestelde architectuur werkt. Bovendien presenteren we simulatie- en experimentele resultaten die de haalbaarheid en efficiëntie ervan aantonen. We openen een proof-of-concept-prototype van de voorgestelde architectuur zodat onderzoekers en ontwikkelaars experimenten kunnen uitvoeren of het concept kunnen uitbreiden naar andere toepassingen.

In de afgelopen twee hoofdstukken hebben we effectieve op SDR gebaseerde architecturen en technieken voor radio HW-virtualisatie en softwarisatie gepresenteerd en besproken. Vanaf Hoofdstuk 4 stellen we geavanceerde SDR-architecturen en technieken voor voor efficiënt en eerlijk delen van spectrum. De explosieve opkomst van draadloze / mobiele standaarden, zowel voor gelicentieerde als niet-gelicentieerde spectrumbanden, heeft geleid tot de creatie van een enorme hoeveelheid technologieën, waarvan er vele strijden om dezelfde spectrumband in plaats van deze harmonieus te delen. Helaas is het radiospectrum een schaarse hulpbron en zullen de beschikbare frequentiebanden niet meeschalen met de verwachte vraag. Als zodanig is het dringend nodig om geavanceerde architecturen en technieken te ontwikkelen om intelligent en eerlijk toegang te krijgen tot het spectrum en het te delen. Daarom stellen we in Hoofdstuk 4 een nieuw open-source en runtime configureerbaar SDR-gebaseerd raamwerk voor spectrumdeling voor dat kan worden gebruikt om het gebruik van het radiospectrum te maximaliseren. Dit raamwerk bestaat uit drie hoofdmodules: (i) op slot gebaseerde PHY-laag (voor discontinue datatransmissie en -ontvangst), (ii) RF-monitor (voor spectrumdetectiedoeleinden) en (iii) LBT (voor RAT-coëxistentie). In dit hoofdstuk bespreken we drie gevallen waarin het voorgestelde kader kan worden gebruikt in combinatie met intelligente algoritmen om het gebruik van het spectrum te maximaliseren. Om het hoofdstuk af te sluiten, bieden en bespreken we experimentele resultaten die met het raamwerk zijn uitgevoerd.

Voortbordurend op het onderzoek naar architecturen en technieken voor efficiënt en eerlijk delen van spectrum, presenteren we in hoofdstuk 5 een open-source, sterk geoptimaliseerde en real-time configureerbare SDR-gebaseerde PHY-laag voor onderzoek en ontwikkeling van nieuwe intelligente spectrumdeling. technieken en algoritmen. Deze PHY-laag is gebouwd op basis van het raamwerk dat in het vorige hoofdstuk is geïntroduceerd en voegt verschillende verbeteringen toe aan de bovenkant. Deze PHY-laag heette SCATTER PHY en maakte deel uit van het SCATTER-systeem, dat is ontworpen en gebouwd als een meerlagig en ontkoppeld communicatiesysteem, met PHY, MAC, AI en andere lagen die allemaal zijn verbonden via een ZeroMQ-bus, waardoor de lagen om te communiceren via een reeks goed gedefinieerde berichten. Het SCATTER-systeem is ontworpen om deel uit te maken van de DARPA Spectrum Collaboration Challenge, die als primair doel had het onderzoek en de ontwikkeling van intelligente coëxistentie- en samenwerkingstechnieken van heterogene netwerken die in dezelfde spectrumbanden opereren. Dit hoofdstuk bespreekt het ontwerp en de belangrijkste kenmerken van SCATTER PHY en sluit af met de resultaten van experimenten die zijn uitgevoerd om de effectiviteit en prestaties te beoordelen.

Samenvattend, het werk dat in dit proefschrift wordt uitgevoerd, behandelt verschillende onderzoeken en oplossingen voor radio HW virtualisatie / softwarisatie en efficiënt spectrum delen. Het onderzoek heeft betrekking op verschillende belangrijke onderwerpen, waaronder: de virtualisatie en softwarisatie van RF-front-ends, SDWN's, multi-wireless-standard en multi-channel enabled architecturen, DSS, C-RAN en intelligente en eerlijkere spectrum sharing-oplossingen. Dit onderzoek heeft geresulteerd in verschillende publicaties die zijn gepubliceerd in respectievelijk relevante internationale tijdschriften en conferenties. Dit doctoraat levert substantiële bijdragen aan zijn domein en dat kan worden gebruikt als basis voor het onderzoek en de ontwikkeling van de volgende generatie draadloze netwerken.

Summary

The availability of ubiquitous wireless and mobile networks, the emergence of new technologies and standards, covering different frequency ranges either licensed and unlicensed has triggered the appearance of a myriad of wireless and mobile devices. Additionally, it is expected that soon, the number of connected devices will largely exceed the number of humans. Due to this exceptional growth, the radio spectrum is getting more and more saturated leading to sub-optimal spectrum usage. Many of these devices co-exist in the same shared wireless environment and unfortunately, the radio spectrum is a scarce resource where the available spectral bandwidth does not scale with the requested wireless bandwidth. Some chunks of the wireless radio spectrum are overcrowded (e.g., license-free ISM bands) while others might be significantly underutilized (TV and FM licensed bands). Therefore, there is an urgent need to come up with new architectures and techniques that can be used to access and share the radio spectrum in a more intelligent and fair way. These architectures and techniques should aim at improving the coordination within a given spectral band or exploring the possibilities of opportunistic spectrum use of unused spectrum in underutilized frequency bands. Therefore, to deal with improved coordination and fairer spectrum sharing, advanced SDR architectures and techniques have been introduced for radio HW virtualization and for accessing and sharing the radio spectrum in a highly efficient, fair and effective way.

The first part of this dissertation, which is composed by Chapter 1, situates the conducted research work, summarizes the main challenges and contributions and outlines the structure of this dissertation. It also provides an overview of the publications that were authored during this research period. In this chapter, we initially present the reader with some context on the topics covered in this work. First, we present SDWN, which can be thought of as the application of SDN concepts in wireless and mobile networks, and resource slicing, which is a technique that creates separate and independent traffic classes into separate sub-spaces. In the sequence, we also talk about C-RAN and functional split, which are concepts that are used to achieve dynamic shared resource allocation in RANs. We also discuss the implementation constraints involved in deploying C-RANs. Next, we provide some context on the second topic covered by this work, which is spectrum sharing. For this research topic, we talk about the scarcity and underutilization of the radio spectrum we face today that is brought about by tight regulation of the spectrum and the huge number of mobile and wireless devices fighting for an opportunity to transmit. Next, we talk about cognitive and intelligent radios, which are foreseen

to be technology that will optimize the scarce and underutilized radio spectrum. Finally, we introduce DARPA's Spectrum Collaboration Challenge, which aimed at having its participants devising novel spectrum access strategies and paradigms in which radio networks will autonomously collaborate and reason about how to dynamically share the radio spectrum. In the sequence of this chapter, based on what was discussed previously, we introduce the overall challenges tackled in this research. The first two challenges are RAN virtualization and spectrum virtualization, which discusses the difficulties involved in virtualizing and softwarizing networks resources including physical RF front-ends. Next, we show that the virtualization and softwarization of RF front-ends can address the challenges created by DSS. The remaining challenges cover the spectrum sharing research topic, which was also covered by this PhD work. We start by discussing the challenges involved in having fairer spectrum sharing and end by talking about the possible challenges that can be addressed by what we refer to as intelligent spectrum sharing. In intelligent spectrum sharing, AI-based radios and networks are used to maximize the spectrum usage. To conclude the chapter, we describe how each remaining chapter of this dissertation is composed of publications executed within the scope of this PhD, list the research contributions that were achieved and made available by those publications, and the list of research results obtained during this PhD research.

Current SDWN implementations mainly use customized firmware, which runs on commercial embedded application processor chips, and are restricted to layers above lower MAC. This limitation comes from the fact that radio HW usually requires specific drivers, which have proprietary implementations. Thus, it is quite complex, if not impossible, to achieve virtualization and softwarization of the radio HW in those cases. Virtualization and softwarization are key-technology enablers for accommodating the growing demand of 5G data services, therefore, in Chapter 2, we investigate the necessity and feasibility of extending the virtualization and softwarization of wireless networks towards the radio HW. We propose a virtualized and softwarized radio HW transceiver architecture, which brings softwarization and virtualization techniques down to the PHY and MAC layers. Additionally, we discuss recent progresses in flexible and generic PHY layer implementations, the trend towards dynamic spectrum allocation schemes, and real-life SDWN practices. Besides that, we present ORCA's project vision and goals. Finally, we assess latency requirements and present a proof-of-concept demonstration of the proposed architecture.

The next generation of wireless and mobile networks will have to face several challenges ranging from a huge increase of traffic loads and efficient use of network and spectrum resources. This scenario calls for novel ways to maximize the both resources (i.e., network and spectrum). Therefore, in Chapter 3, we propose a SW-based hypervisor architecture that efficiently abstracts a RF front-end into a configurable number of virtual RF front-ends. It enables flexible and dynamic spectrum access and sharing in existing wireless and mobile networks. This is a highly-optimized and non-intrusive hypervisor architecture that multiplexes the signals of several different and concurrent multi-carrier-based RATs with correlated numerologies. By employing the proposed architecture it is possible to reduce the costs and increase the spectral efficiency through densification. To make this chapter complete, we provide mathematical demonstration how the proposed architecture works. Additionally, we present simulation and experimental results attesting its feasibility and efficiency. We provide a proof-of-concept prototype of the proposed architecture as an open-source framework for researchers and developers to run experiments or extend the concept to other applications.

In the past two chapters, we presented and discussed effective SDR-based architectures and techniques for radio HW virtualization and softwarization. From Chapter 4 onward, we propose advanced SDR architectures and techniques for efficient and fair spectrum sharing. The explosive emergence of wireless/mobile standards, covering licensed and unlicensed spectrum bands brought about the creation of a huge amount of technologies, with many of them competing for the same spectrum band instead of harmoniously sharing it. Unfortunately, the radio spectrum is a scarce resource, and the available frequency bands will not scale with the foreseen demand. As such, there is a urgent need to devise advanced architectures and techniques to intelligently and fairly access and share the spectrum. Therefore, in Chapter 4, we propose a novel open-source and runtime configurable SDR-based spectrum sharing framework that can be employed to maximize the usage of the radio spectrum. This framework is composed of three main modules: (i) slot-based PHY layer (for discontinuous data transmission and reception), (ii) RF-Monitor (for spectrum sensing purposes) and (iii) LBT (for RAT coexistence). Within this chapter, we discuss three import use cases where the proposed framework can be used along with intelligent algorithms to maximize the spectrum's usage. To conclude the chapter, we provide and discuss experimental results performed with the framework.

Continuing with the research on architectures and techniques for efficient and fair spectrum sharing, in Chapter 5, we present an open-source, highly-optimised, and real-time configurable SDR-based PHY layer for the research and development of novel intelligent spectrum sharing techniques and algorithms. This PHY layer is built based on the framework introduced in the previous chapter and adds several improvements on its top. This PHY layer was named SCATTER PHY and was part of the SCATTER system, which was designed and built as a multilayered and decoupled communications system, with PHY, MAC, AI, and other layers all connected through a ZeroMQ bus, which allows the layers to communicate through a set of well defined messages. The SCATTER system was designed to be part of the DARPA Spectrum Collaboration Challenge, which had as primary goal the research and development of intelligent coexistence and collaboration techniques of heterogeneous networks operating in the same spectrum bands. The chapter discusses the design and main features of SCATTER PHY and concludes with the results of experiments performed to assess its effectiveness and performance.

In summary, the work conducted within this dissertation deals with several studies and solutions for radio HW virtualization/softwarization and efficient spectrum sharing. The research covers several import of topics, including: the virtual-

ization and softwarization of RF front-ends, SDWNs, multi-wireless-standard and multi-channel enabled architectures, DSS, C-RAN, and intelligent and fairer spectrum sharing solutions. This research resulted in several publications that have been published in relevant international journals and conferences, respectively. This PhD provides substantial contributions to its domain and that can be employed as basis for the research and development of the next generation of wireless networks.

1
Introduction

"Go back? No good at all. Go sideways? Impossible. Go forward? Only thing to do. On we go."

– J. R. R. Tolkien, The Hobbit (1892 - 1973)

During the past decades, we have witnessed the explosive emergence of wireless technologies and standards, covering different ranges (from short-range low-power personal, local area up to wide area networks) and different licensed and unlicensed radio spectral bands (from sub 6 GHz up to 60 GHz). This explosion has been accompanied by an huge demand increase in data rates and number of connections. The report in [1] predicts that data rates will easily reach peaks of 10 Gigabits per second (Gbps), reaching a staggering 49 Exabytes of data transfer per month. There will be 29.3 billion connected devices by 2023, including Machine to Machine (M2M) modules. The report states that future mobile networks will need to support 1 million connections per square kilometer and up to a total of 100 billion connections in total.

Based on these changes and demands, some observed trends are: (i) an ever-increasing number and diversity of wireless devices (ii) the execution of critical and bandwidth-hungry applications with very diverging Quality of Service (QoS) requirements (e.g., in terms of throughput, latency, reliability, energy consumption, synchronization, coverage, etc.), and (iii) the presence of multiple heterogeneous wireless technologies/standards, often integrated into the same device, sharing the same physical environment and in many cases also sharing the same spectral bands.

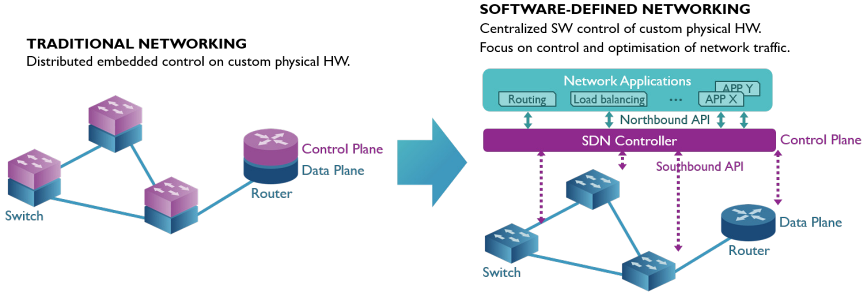


Figure 1.1: Architectural Concept of SDN.

Therefore, to cope with these demanding trends, novel and advanced Software-Defined Radio (SDR) architectures and techniques must be introduced for radio Hardware (HW) virtualization and for accessing and sharing the wireless medium in a highly efficient, fair and effective manner.

1.1 Context

This section situates the conducted research work, giving some context to the reader.

1.1.1 Software-Defined Wireless Networks

This subsection presents the reader with some context on Software-Defined Wireless Networks.

1.1.1.1 Software-Defined Wireless Networks

Software-Defined Network (SDN) is a network paradigm that removes the control functions from the equipment, thus creating independent and decoupled control and data planes. The control plane is responsible for deciding what to do with each data flow in the network and the decisions taken by the control plane are enforced in the equipment of the data plane. This separation of functions enables direct programmability of network controls and an abstraction of the underlying infrastructure for wireless applications. This programmability, for example, removes the need to manually configuring each piece of equipment in the network [2]. With SDN, it is possible to create a service delivery platform that is adaptable to the users' varying demands. The architectural concept behind the SDN technology is illustrated in 1.1. The figure shows the evolution from traditional networking to SDN.

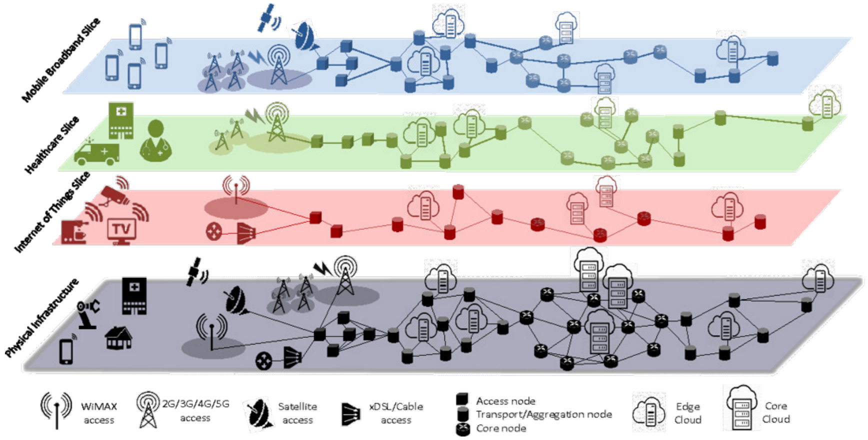


Figure 1.2: Network virtualization.

The growing popularity of smart phones, tablet computers and mobile cloud services places an increasing demand for dynamic services from wireless networks. This demand creates new requirements for the network architecture, such as flexibility in management and configuration, adaptability and vendor independence. To meet these requirements, Software-Defined Wireless Network (SDWN) has been proposed as a cost-effective solution. Therefore, SDWNs can be understood as the use of SDN concepts in wireless networks. By using a controller in the control plane, SDWN facilitates the creation of new adaptive mechanisms according to different applications and user demands, such as mobility (handoff/handover), latency, security, and QoS [3]. However, issues such as supporting a large number of subscribers, frequent mobility, fine-grained measurement and control, low-latency, and real-time adaptation need to be addressed by future SDWN architectures.

1.1.1.2 Network Resource Slicing

Network resource slicing or just slicing is a technique that separates traffic classes, i.e., a bunch of flows having similar traffic characteristics and QoS needs, into separate sub-spaces. Generally, slicing is done for supporting services with a specific vertical, e.g. a slice for mission-critical communication, a slice for industry-automation in a factory environment, etc. These sub-spaces, or slices, are managed by a controller and share network resources [4]. Much like virtual memory for processes in an operating system, each slice creates the illusion of its own distinct network resources. In reality, many slices can be running on the same HW. When combined with per-class rules, this abstraction can isolate different slices of the network. Slices can be easily customized and create, on top of a common physical

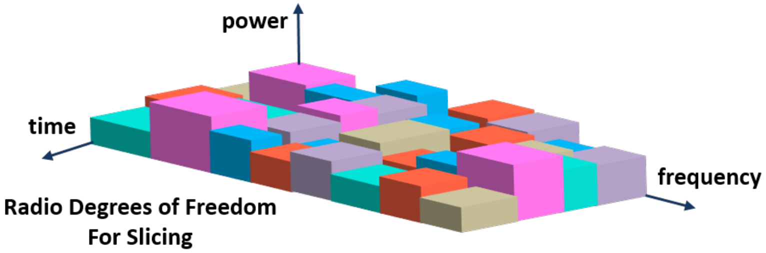


Figure 1.3: An example of spectrum-level slicing.

infrastructure, a set of logical networks, each tailored to accommodate fine-tuned Service Level Agreements (SLAs) (i.e., service types) reflecting the needs of different services. Figure 1.2 depicts the sharing of physical network resources by the creation of isolated virtual networks, i.e., network slices. Each network slice can be individually configured to serve a particular purpose (vertical), guaranteeing a particular set of performance characteristics.

Slicing will be key enabler for addressing the issue of rapidly growing demand in mobile data services related to 5th Generation (5G) cellular networks and to unlocking 5G revenue opportunities such as enhanced video, and in-car connectivity and extended reality.

One aspect of the definition of a slice is related to where in the network or up to what level slicing should be applied. A suitable classification of this facet of slicing is given in [5], where different levels of slicing are detailed:

(i) **Spectrum-level slicing**: This kind of slicing can be thought of being an extension of Dynamic Spectrum Sharing (DSS) and spectrum sharing. The spectrum can be sliced by multiplexing time, space, frequency or by a combination of those. It can be considered as link virtualization. It is illustrated by Figure 1.3;

(ii) **Infrastructure-level slicing**: It is the slicing of physical network elements, such as antennas, base stations (BSs), processors, memory, routers, etc. These elements are virtualized to support sharing by multiple operators. It is accomplished mostly by virtualization. Figure 1.4 shows an example of infrastructure-slicing;

(iii) **Network-level slicing**: It is the slicing of all the network infrastructure. From a pure network virtualization aspect, the basic difference between network and infrastructure slicing is that with network slicing, operators can manage the virtualized infrastructure and virtual machines sufficiently through the virtualization controllers. Network-level slicing is considered

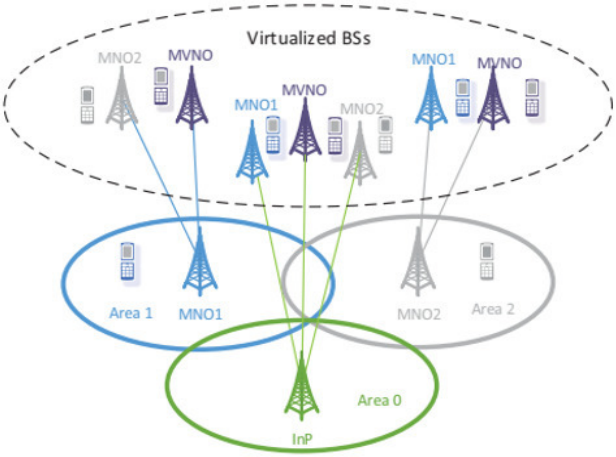


Figure 1.4: An example of infrastructure-level slicing [5].

the ideal case for virtualization. An example of network-level slicing is depicted in Figure 1.5.

1.1.1.3 C-RAN

A Radio Access Network (RAN) is the part of the mobile telecommunication system that implements a Radio Access Technologies (RAT). A RAN consists of of many stand-alone BSs that provide access to user devices. Each BS covers a small area, whereas a group of BSs provides coverage over a continuous area. Each BS has its own cooling, backhaul transportation, backup battery, monitoring system, and so on. Conceptually, a RAN provides connection between a device (e.g., a mobile phone) and the core network. In addition to these functionalities, it also performs essential management and resource allocation functionalities [6].

There are several limitations in the traditional cellular architecture. First, each BS is costly to build and operate. Second, when more BSs are added to a system to improve its capacity, interference among BSs is more severe as BSs are closer to each other and more of them are using the same frequency. Third, because users are mobile, the traffic of each BSs fluctuates (it is known as the 'tide effect'), and as a result, the average utilization rate of individual BSs is pretty low. However, the BS's processing resources cannot be shared with other BSs. Therefore, all BSs are designed to handle the maximum traffic, not average traffic, resulting in a waste of processing resources and power at idle times.

In order to tackle those issues, a distributed BS architecture was introduced by some telecommunications equipment vendors [9]. In this architecture the radio function unit, also known as the Remote Radio Head (RRH), is separated from the

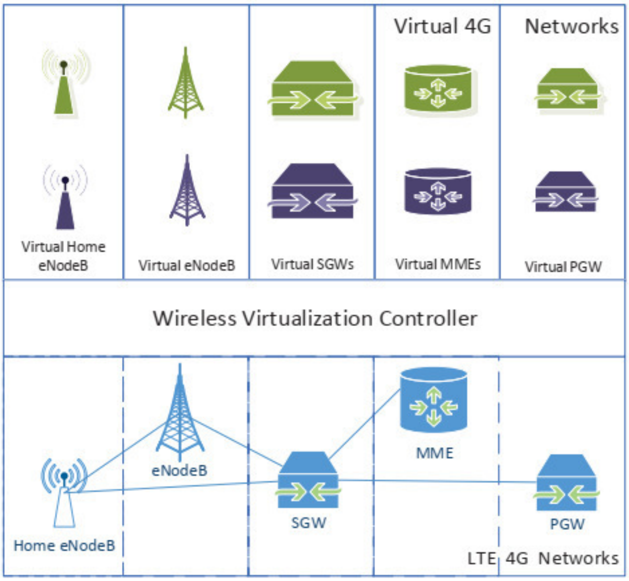
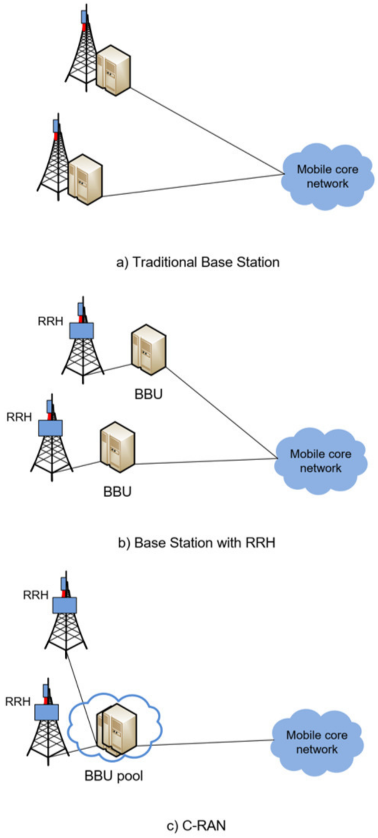


Figure 1.5: An example of network-level slicing [5].

digital function unit, or Baseband Unit (BBU) by optical fiber. The RRH can be installed on the top of a tower close to the antenna, reducing the loss compared to the traditional BS where the Radio Frequency (RF) signal has to travel through a long cable from the BS cabinet to the antenna at the top of the tower. The optical fiber link between RRH and BBU also allows more flexibility in network planning and deployment as they can be placed a few hundreds meters or a few kilometers away. Figure 1.6 illustrates the evolution of the RAN architecture, it shows the difference between traditional RAN and Cloud/Centralized Radio Access Networks (C-RAN).

C-RAN may be viewed as an architectural evolution of distributed BS system. It takes advantage of many technological advances in wireless, optical and IT communications systems. It applies recent Data Center Network technology to allow a low cost, high reliability, low latency and high bandwidth interconnect network in a pool of BBUs. It utilizes open platforms and real-time virtualization technology rooted in cloud computing to achieve dynamic shared resource allocation and support multi-vendor, multi-technology environments [10]. This way, the heavy processing functions can be run on general purpose HW, and therefore architectural solution reduces capital costs and promotes the deployment of new technologies.



a) Traditional Base Station

b) Base Station with RRH

c) C-RAN

Figure 1.6: Evolution of the RAN architecture [11].

1.1.1.4 Functional Split

Open radio access networks offer the option of placing network functions in different places along the signal path. That option, called a functional split, lets network engineers optimize performance and make trade-offs. The functional split determines how many base station functions to leave locally, close to the user, with the benefit of relaxing fronthaul network bitrate and latency requirements, and how many functions to centralize with the possibility of achieving greater processing benefits.

Figure 1.7 shows some possible functional splits. Traditional wireless systems (as shown at the right), have the lower layer radio functionality integrated with RF-frontend and antenna-system and only higher layer networking functions are centrally controlled. In the extreme softwarization case (shown at the left), the the entire protocol stack is centrally controlled in the cloud, and the digital baseband

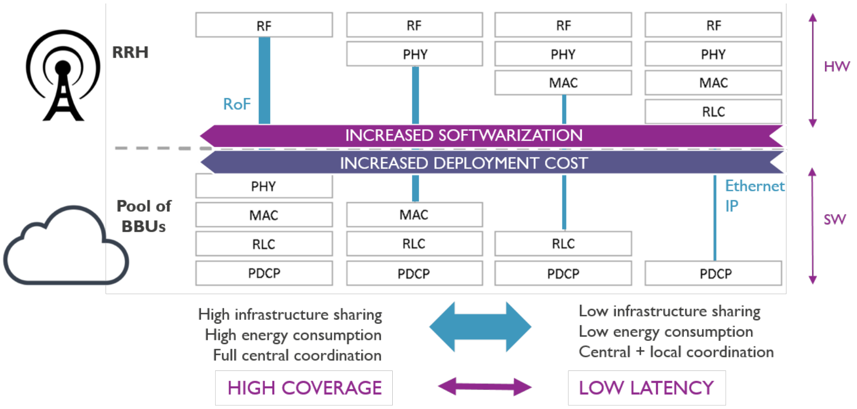


Figure 1.7: Functional splitting.

signal is sent to a distributed unit, called RRH. Please note that digital based bitrate is typically two orders of magnitude higher than the actual user date rate carried by the baseband signal.

Despite its appeal, one key obstacle in the adoption of C-RAN is the excessive capacity requirements on the fronthaul link that provides connection between the pool of BBUs and the RRHs. To relax the excessive fronthaul requirement, the concept of C-RAN is being revisited, and a more flexible distribution of baseband functionalities between the RRH and BBU is being considered [12]. Rather than offloading all baseband processing to the BBU, it is possible to keep a subset of these blocks in the RRH. This concept is known as functional split. By gradually splitting and placing increasingly more baseband processing functions at RRHs, the fronthaul capacity requirement reduces considerably [13].

Nevertheless, flexible centralization requires more complex RRHs and reduces the opportunities to perform coordinated processing and advanced interference avoidance. Consequently, the flexible centralization is a trade-off between what is gained in terms of relaxing the fronthaul performance requirements, and what is lost in terms of perceived user performance under few coordinated processing that was expected in original C-RAN architecture [14].

1.1.1.5 Implementation Constraints of C-RAN

Software (SW) implementation of RAN functionality requires a new way to design and operate the RAN. Until now, RAN functionality has been executed on dedicated and not shared HW such as Digital Signal Processors (DSPs) or Application-Specific Integrated Circuits (ASICs). Dedicated and not shared HW is precisely dimensioned and provides the required resources to cope with peak-traffic demands; it is highly reliable and has high performance, but does not permit resource shar-

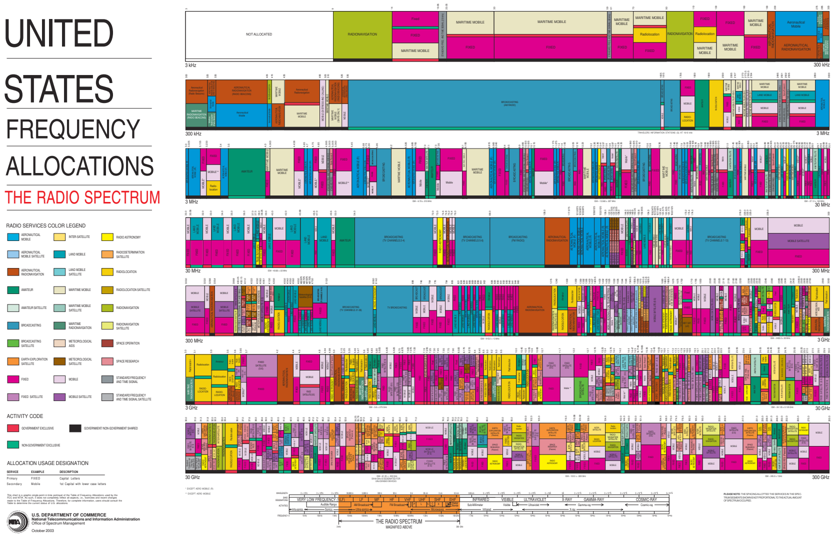


Figure 1.8: Frequency allocation in the United States [17].

ing or virtualization. In contrast, SW implementations on commodity HW may be more flexible and allow for resource sharing and virtualization. However, it is usually less reliable and has lower performance. Therefore, such implementations need to be ”cloud-native” and must be designed for resilience. This cannot be achieved by merely porting existing implementations, but rather requires more advanced concepts [15].

C-RAN will pose new challenges to data-center architectures since it may require dedicated platforms rather than the existing platforms that have been optimized for Internet services. However, they will still be considered ”commodity” due to the pervasiveness of mobile network technology. In particular, the distribution and execution of RAN processing jobs in data centers requires high-performance SDN architectures that route RAN data and address processing elements within data centers efficiently [15]. Similarly, the real-time requirements in a RAN may not allow simple migration of virtual machines (or containers) but require new mechanisms that facilitate fast transfer of processing states or RAN protocol states. The efficiency with which processing elements (containers or virtual machines) are assigned to data packets has a major impact on the elasticity of the system.

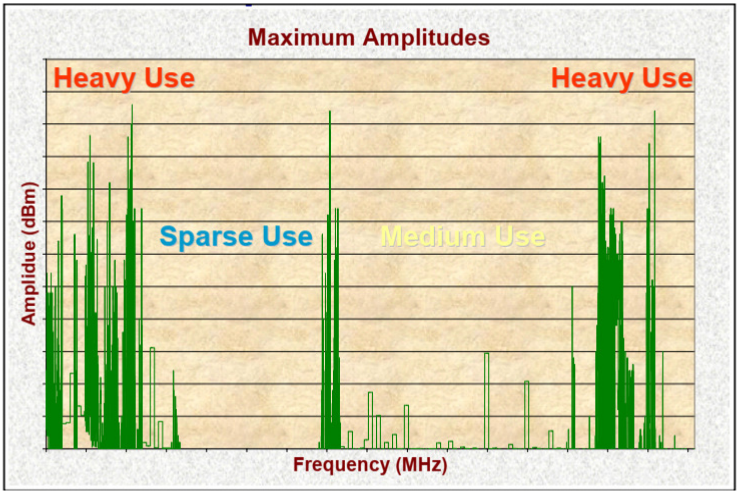


Figure 1.9: Example of spectrum underutilization [17].

1.1.2 Spectrum Issues

This subsection presents the reader with context on the spectrum's (i) scarcity, (ii) current fixed allocation of bands, and (iii) sharing strategies.

1.1.2.1 The Scarcity and Underutilization of the Radio Spectrum

Radio spectrum is one of the most tightly regulated resources of all time. From cellphones to police scanners, from Television (TV) sets to garage-door openers, virtually every wireless device depends on access to the radio frequency wireless spectrum. However, access to spectrum has been quite limited ever since RF transmissions were first regulated in the early 20th century. Figure 1.8 shows the frequency allocation in the the U.S., where we can see the scarcity of free spectrum bands for new technologies and services. The figure shows that the wireless world today is a patchwork of planned isolation. As can be seen, static allocation is an inefficient solution to ever-increasing demand for spectrum. However, recent studies show that most of the assigned spectrum is underutilized [16, 17]. Figure 1.9 shows the signal strength distribution over a large portion of the wireless spectrum in the U.S. [17]. It illustrates that a large portion of the assigned spectrum is used sporadically or never used at all. The spectrum usage is concentrated on certain portions of the spectrum while a significant amount of the spectrum remains underutilized.

On the other hand, the increasing number of wireless technologies and services leads to a spectrum scarcity [18]. Spectrum scarcity has emerged as a primary problem encountered when trying to launch new wireless services. The effects of this scarcity are most noticeable in spectrum auctions where the operators often need to invest billions of dollars to secure access to specified bands in the available spectrum. In spite of this scarcity, recent spectrum utilization measurements have shown that the available spectrum opportunities are severely underutilized, i.e., left unused [16].

For instance, a study conducted by the Federal Communications Commission (FCC) and reported in [19], showed that usage of the licensed radio spectrum varied from 15 % to 85 % in the United States of America. This means that the primary users (also known as licensed or legacy users) do not often occupy the allocated radio resources incessantly and this leads to their underutilization. Also, as reported in [19], similar actual spectral occupancy measurements showed that as little as 22 % of allocated spectrum is utilized in urban areas and less than 3 % is being utilized in rural areas, which highlights the fact that usage of radio spectrum varies with location. Similarly, as reported in [19, 20], observation from actual radio spectrum occupancy measurements conducted in downtown Berkeley showed that allocated frequency bands to licensed users are underutilized especially in the 3-10 GHz, while some other frequency bands are rarely occupied and the remaining frequency bands are heavily occupied showing that radio spectrum usage also varies with frequency.

1.1.2.2 Cognitive and Intelligent Radios

The availability of ubiquitous wireless networks, the emergence of new technologies and standards, covering different frequency ranges either licensed and unlicensed has triggered the appearance of a myriad of wireless and mobile devices. Many of these devices co-exist in the same environment and unfortunately, the radio spectrum is a scarce resource where the available bandwidth does not scale with the requested wireless bandwidth. Some chunks of the wireless radio spectrum are overcrowded (e.g., license-free Industrial, Scientific, and Medical (ISM) bands) while others might be significantly underutilized (TV and Frequency Modulation (FM) licensed bands). Therefore, there is an urgent need to come up with new techniques that can be used to access and share the spectrum in a more intelligent way. These techniques should aim at improving the coordination within a given spectral band or exploring the possibilities of opportunistic spectrum use of unused spectrum in underutilized frequency bands. To improve or optimize this sub-optimal use of the radio spectrum that exists today, cognitive and intelligent SDRs seek to automatically reconfigure their transmission, reception, and network parameters, adapting the radio to the environment in which it is operating. These radios can be programmed and configured dynamically to use the best wireless channels in their vicinity to avoid user interference and congestion. Such a radio automatically detects available channels in the wireless spectrum, then accordingly changes the proper parameters to allow more concurrent wireless communications

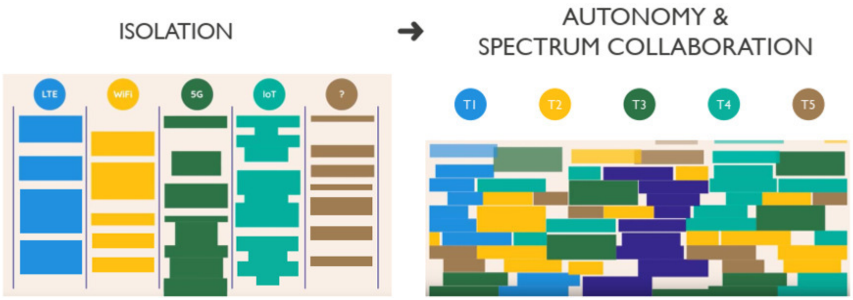


Figure 1.10: Moving collaboratively away from spectrum isolation [1].

in a given spectrum band at one location.

Such radios are expected to either improve the coordination within a given spectral band (horizontal spectrum sharing) or explore the possibilities of opportunistically using unused spectrum in underutilized and potentially licensed spectral bands (vertical spectrum sharing, also known as DSS).

1.1.2.3 DARPA’s Spectrum Collaboration Challenge

The wireless revolution is fuelling a voracious demand for access to the RF spectrum worldwide. In the civilian sector, consumer devices, from smartphones to wearable fitness recorders to smart kitchen appliances, compete for bandwidth. There is a growing reliance on unmanned platforms in the military sector, from underwater sensors to satellites, and a push for broadband connectivity for every member of every Service. Managing this increasing demand while combating what appears to be a looming scarcity of RF spectrum is a severe problem for all nations.

Today’s approach, which is nearly a century old, isolates wireless systems by dividing the spectrum into rigid, exclusively licensed bands allocated over large, geographically defined regions. This approach rations access to the spectrum in exchange for the guarantee of interference-free communication. However, it is based on human-defined rules for spectrum access and not adaptive to supply and demand dynamics. So it cannot exploit the total potential capacity of the spectrum. At any given time, many allocated bands are unused by licensees while other bands are overwhelmed, thus squandering the spectrum’s enormous capacity and unnecessarily creating conditions of scarcity.

The Spectrum Collaboration Challenge (SC2) was created to ensure that the exponentially growing number of military and civilian wireless devices will have full access to the increasingly crowded electromagnetic spectrum. During the competition, competitors were asked to re-imagine spectrum access strategies and de-

[1] Source: Darpa SC2, https://youtu.be/cd3kCPvaXOw

velop a new wireless paradigm in which radio networks will autonomously collaborate and reason about how to dynamically and collaboratively share the RF spectrum, thereby avoiding interference and jointly exploiting opportunities to achieve the most efficient use of the available spectrum. SC2 teams developed breakthrough capabilities by taking advantage of recent advances in Artificial Intelligence (AI) and machine learning (ML), and the expanding capacities of SDRs. The competition aimed to produce breakthroughs in collaborative AI and to catalyze a new paradigm for spectrum use that can help usher in an era of spectrum abundance [21]. Figure 1.10 illustrates a move from the case where isolated spectrum silos lead to over dimensioning and waste of spectrum to the point where autonomous and collaboratively spectrum usage is achieved by adding intelligence to the networks.

1.2 Challenges

The efficient virtualization and softwarization of radio HW (i.e., the RF front-end) resources is of utmost importance to the next-generation of wireless and mobile networks. Additionally, dynamic spectrum access requires highly efficient and preferably low-cost solutions to monitor and access the radio spectrum. The overall challenges tackled in this research are listed below.

1.2.1 RAN Virtualization

RAN virtualization can be performed at different levels, based on the type of resource that is being sliced: spectrum, access-technology, or link virtualization. Similarly to wired network virtualization, in which the physical device is sliced and abstracted into multiple virtual counterparts, RAN virtualization needs physical wireless devices and radio resources to be sliced and abstracted into a number of virtual counterparts. In other words, virtualization, in both wired and wireless networks, can be considered as the process of splitting the entire network system [5, 22]. However, the distinctive properties of the wireless environment, e.g., attenuation, mobility, and broadcast, make the process of slicing and abstraction more complicated. Furthermore, RANs can provide connectivity services to a much wider range of access technologies than wired networks, which make the abstraction challenging to achieve.

In RAN virtualization, a resource can refer to the wireless equipment such as the entire BS, as well as low-level physical resources such as space-time-frequency slots. The abstraction of low-level resources provides the illusion that high-level resources (such as entire BSs) are being virtualized, although low-level resources must ultimately be partitioned to support that sharing.

1.2.2 Spectrum Virtualization

Virtualization allows a single physical HW to implement multiple instances of a required logical resource within the same or different slices. However, wireless networks pose several unique challenges that make ”slicing” different for wireless networks compared to wired networks. First, some novel applications will focus on new Physical (PHY) and Medium Access Control (MAC) layer protocols, and hence sharing a single ”physical” wireless node between multiple such applications becomes quite hard if not impossible. Second, mobility and handover procedures typically have very tight latency requirements, which also become extremely difficult, if not impossible, to support on a network where the applications get access to only a few resources out of the whole set of physical HW in the topology of interest [23].

Next-generation wireless and mobile networks are expected to handle a large number of devices and data intensive applications with different requirements (e.g., low latency, high security, high data rate, and optimal bandwidth). One type of network cannot serve such diverse requirements. Spectrum virtualization is regarded as an emerging technology for future wireless networks to enhance both RF spectrum utilization and overall network performance while supporting connectivity for massive number of devices. Spectrum virtualization involves the abstraction and sharing of physical resources.

Virtualization at the spectrum level consists of abstracting and sharing the radio spectrum through frequency, time, space, code dimensions, or a combination of those dimensions. It was was first proposed in [23]. In that work, the authors conceptually discuss several methods to achieve spectrum abstraction, all of them based on the multiplexing of time, frequency, and space resources. For instance, spectrum virtualization can used to enable multiple concurrent RATs to run on top of the same set of radio devices.

With spectrum virtualization, the physical RF front-end is virtualized by a layer of virtualization (i.e., a hypervisor) that creates a virtual radio resource pool. In other words, the same physical device can host different functions that belong to various virtual networks increasing separation from the underlying HW [24]. The virtualization of spectrum resources has the potential to enhance spectrum efficiency, energy efficiency, coverage, latency, data rate, and QoS in wireless and mobile networks [25]. Figure 1.11 depicts the virtualization of a physical RF front-end, which enables several virtual networks to run onto several virtual front-ends.

1.2.3 Dynamic Spectrum Access

Traditional wireless networks work on a fixed set of predefined channels. This, however, becomes inefficient as the frequency bands are shared by more and more heterogeneous wireless networks. For instance, in the 2.4 GHz ISM band, trans-

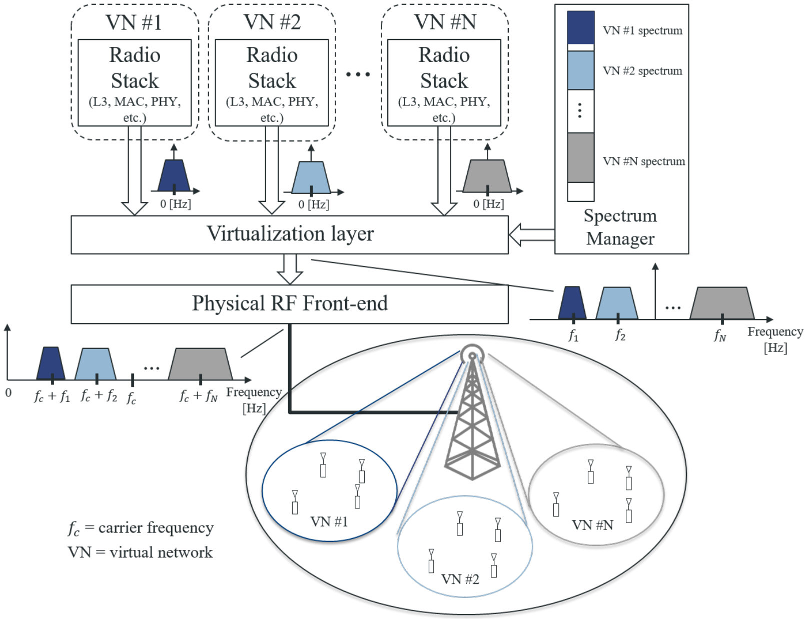


Figure 1.11: Virtualization of a physical RF front-end.

missions of a narrow band wireless network (e.g., Internet of Things (IoT) devices using IEEE 802.15.4 technologies) may cause interference with a wide-band wireless network (e.g., devices using 802.11 technologies) and cause it to back-off, wasting a large portion of wireless frequency. It is essential to mention that due to the enormous power difference between IoT and 802.11 (Wi-Fi) devices, only IoT devices sufficiently close to a Wi-Fi device will cause interference. However, the impact of Wi-Fi devices on IoT devices is more significant.

To improve the spectrum efficiency, next-generation devices should support Dynamic Spectrum Access (DSA) [26]. Unlike traditional wireless devices that operate on fixed spectrum assignments, DSA requires them to be dynamically reconfigurable in a wide spectrum range according to the network or environmental status. DSA has the potential to solve the existing spectrum inefficiency problem [26]. One example of DSA is white-space networking, where wireless devices opportunistically use spare TV channels for data communication [27]. Another important example is Citizens Broadband Radio Service (CBRS), where Long-Term Evolution (LTE) small cells are sharing spectrum in the 3.5 GHz band with naval radar in the U.S. [28]. With DSA, instead of contending with narrow-band devices over a large contiguous channel, the wide-band devices can utilize the available spectrum segments left unused by the narrow-band devices, avoiding mutual interference.

While DSA is desirable, enabling it in existing wireless network designs remains a challenging task due to the lack of spectrum programmability, i.e., the ability to change spectrum properties of a signal to match an arbitrary frequency allocation. First, conventional wireless standards are primarily designed for static and monolithic spectra. Therefore, they have minimal spectrum, and bandwidth agility [29]. DSA discards the widely adopted design principle of having fixed channels. Instead, each device must now handle more adaptive time/space varying spectrum allocation, including changing frequency, varying width, and non-contiguous bands [30]. That poses a substantial challenge to any PHY layer design that assumes the traditional fixed spectrum model. Second, future wireless PHY layers may support more flexible spectrum programmability by adopting Non-Contiguous OFDM (NC-OFDM) schemes [31, 32], for example. Further, when multiple wireless designs coexist on a single device, there isn't a good way to coordinate the possibly overlapping spectrum management and access functions implemented separately in each wireless PHY/MAC, especially if they are on the same spectrum band (like Wi-Fi and IEEE 802.15.4 over 2.4 GHz ISM band).

1.2.4 Radio Hardware Hypervisor

The next-generation of wireless and mobile networks is expected to provide connectivity services to a large number of devices with hugely different requirements, ranging from mobile/wireless users with high-Bandwidth (BW) services (e.g., video-streaming) to IoT devices with bursty traffic and low-BW services (e.g., metering). To efficiently support such services future networks should be flexible, providing different air-interfaces for particular users and applications [33], e.g., with BSs being capable of serving LTE users and Narrow Band Internet of Things (NB-IoT) devices. In this context, radio HW virtualization is a promising solution to enable multiple virtual radios, i.e., air-interfaces, to coexist on top of HW [34].

Radio HW virtualization requires the development of a hypervisor [5]. The hypervisor must abstract the RF front-end into a number of virtual RF front-ends, which are accessed by SDRs. As several SDRs generate streams of raw, i.e., baseband, In-Phase and Quadrature (IQ) samples to the virtual RF front-ends, the hypervisor has the task to multiplex them into a single waveform signal that will be transmitted by the physical RF front-end. The implementation of such hypervisor architecture poses several challenges, such as the accommodation of a wide range of access technologies, the addition of more flexibility (i.e., programmability of software and hardware) into the radio HW virtualization in order to make it a more suitable solution for 5G networks, etc. Figure 1.12 illustrates a hypervisor multiplexing several baseband IQ data streams into a single wide-band signal (still centered around 0 Hz) that is sent to the physical front-end and then translated into a pass-band signal centered around the carrier frequency.

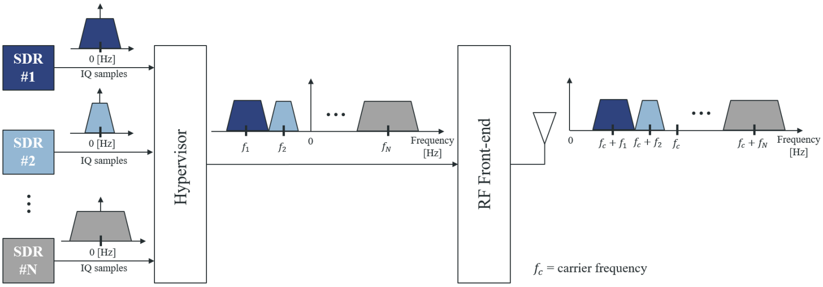


Figure 1.12: Hypervisor multiplexing several SDR streams.

1.2.5 Fair Spectrum Sharing

The dramatic growth in demand for wireless services has fueled a severe spectrum shortage, both in licensed and unlicensed bands. One approach for meeting this galloping demand is to allow the coexistence of competing wireless technologies, such as cellular, Wi-Fi, radar, TV, emergency communications, and others. This shared spectrum paradigm poses novel challenges for the efficient and fair resource allocation. Many of these challenges stem from the heterogeneity of the coexisting systems, their scale, and the lack of explicit coordination mechanisms. Whereas some recent efforts have tried to address the coexistence of specific technologies, a comprehensive and general approach to efficiently coordinate spectrum access for heterogeneous systems remains elusive.

One of the main goals of this research is to provide a framework for fair spectrum sharing between several kinds of networks in licensed and unlicensed bands. When sharing the spectrum, networks become vulnerable to interference from other wireless networks competing for the same spectrum. The challenge here is to propose techniques that minimize the impact of the coexistence of several, sometimes disparate, networks while guaranteeing fair sharing of the radio spectrum resources.

1.2.6 Intelligent Spectrum Sharing

Efficient utilization of the radio spectrum is a fundamental issue of wireless and mobile communications. Licensed frequency allocations are overly inflexible and unlicensed bands are overcrowded with uncoordinated networks, resulting in inefficient usage of the radio spectrum. The conflict between the inefficient usage of spectrum and the rapid growth of wireless and mobile services calls for a more flexible and intelligent solution to manage such an important natural resource. By combining the abilities of spectrum awareness, intelligence and radio flexibility, a cognitive and intelligent radio might be able to adapt itself to the changes in its environment. It is foreseen that a large amount of underutilized spectrum might be efficiently used by applying intelligent spectrum sharing techniques.

Recent works from national Institute of standards and technology (NIST) in the U.S. show that the AI-based strategies greatly outperforms the traditional methods on spectrum sensing and sharing [35]. By using the ML and AI techniques, model-based approaches can be transformed into data-driven ones. In this way, spectrum sharing and management becomes more flexible and efficient. The challenges here range from the development of flexible and efficient SDR frameworks to developing the intelligent layers. These flexible and efficient SDR frameworks should efficiently interact and provide upper layers with intelligence, dynamic configurability, and helpful information on the actual state of the radio environment. Intelligent layers will reason on how to effectively share the spectrum with several diverse and sometimes uncoordinated networks. A perfect synergy between an SDR framework and intelligent layers has the potential to enable smart strategies that optimize the usage of wireless spectrum resources in real-time.

1.3 Outline

This dissertation is composed of a number of publications that were realized within the scope of this PhD. The selected publications provide an integral and consistent overview of the work performed. The different research contributions are detailed in Section 1.4 and the complete list of publications that resulted from this work is presented in Section 1.5. Within this section we give an overview of the remainder of this dissertation.

In Chapter 2, we propose a HW-based architecture to support multi-standard multi-channel virtualization. The architecture is designed to meet the requirements of diverse wireless access technologies at the network edge. The proposed architecture, which encompasses a radio HW virtualization oriented transceiver, makes it possible to bring softwarization and virtualization techniques down to the PHY and MAC layers.

Next, in Chapter 3, we introduce a wireless spectrum hypervisor architecture that abstracts a physical RF front-end into a configurable number of virtual RF front-ends. The proposed architecture has the ability to enable flexible spectrum access in existing wireless and mobile networks. The proposed architecture is a non-intrusive and highly optimized wireless hypervisor that multiplexes the signals of several different and concurrent multi-carrier-based radio access technologies.

Then, in Chapter 4, we present an open-source highly-configurable SDR-based framework that can be employed to devise disruptive techniques to optimize the sub-optimal use of radio spectrum that exists today. The proposed framework enables experimental research and prototyping for various next-generation wireless networks spectrum sharing scenarios.

In Chapter 5, we present SCATTER PHY, which is an open-source PHY layer that uses a filtered form of Orthogonal Frequency Division Multiplexing (OFDM) as its waveform and that is an enhancement/extension of the framework introduced in Chapter 4. This chapter describes the design, features, and benefits of SCATTER PHY. It has been developed as a standalone application, with the ability to communicate with higher layers through a set of well defined messages that are exchanged over a ZeroMQ bus.

Finally, in Chapter 6, we conclude the work by summarizing the main results and concluding the performed work.

Table 1.1 shows the challenges that were highlighted in Section 1.2 and indicates which were targeted per chapter. This can assist the reader in browsing this dissertation.

Table 1.1: An overview of the contributions per chapter in this dissertation.

Challenge	Chapter 2	Chapter 3	Chapter 4	Chapter 5
RAN Virtualization	●	●		
Spectrum Virtualization	●	●		
Dynamic Spectrum Access		●	●	●
Radio HW Hypervisor		●		
Fair Spectrum Sharing			●	●
Intelligent Spectrum Sharing			●	●

1.4 Research contributions

In Section 1.2, the research challenges involved in the radio HW virtualization and efficient spectrum sharing were discussed. As showed in Section 1.3, these challenges are tackled in the remainder of this dissertation. The target of this PhD dissertation is to propose and implement novel and advanced SDR architectures and techniques for radio HW virtualization and efficient spectrum sharing. Therefore, next, we present, on a per chapter basis, the research contributions of this dissertation.

In Chapter 2, we introduce a HW-based architecture supporting low latency processing via Field-Programmable Gate Array (FPGA)-based accelerators. Such architecture is flexible enough to support multi-standard multi-channel virtualization. The flexibility is in terms of SW programmability, which, consequently, offers a fast development cycle, as only SW programming is required for creating and manipulating multiple virtual radios. The architecture presents efficient utilization of FPGA resources, as FPGA-based accelerators can be shared by several multiple virtual radios. The proposed architecture meets the requirements of diverse wireless access technologies at the network edge. In order to reduce design time, the

FPGA-based accelerators were implemented making use of High-Level Synthesis (HLS) tools, where the C programming language is used to describe hardware circuits. We also present a discussion on a demonstration that was implemented based on a SDR platform composed of a Xilinx Zynq System-on-Chip (SoC) and an Analog Devices RF front-end.

Next, in Chapter 3, we propose a non-intrusive and highly optimized wireless hypervisor architecture for SDRs that ensures coexistence, isolation, and programmability for multi-carrier-based systems (i.e., OFDM) with numerologies that are multiple integers of one another. We demonstrate mathematically the idea behind the multiplexing of multi-carrier-based signals performed by the proposed hypervisor architecture. Use cases employing the proposed architecture for DSA and network densification are presented and discussed in detail. Moreover, we describe and discuss some related pieces of work on virtualization, contrasting them with the architecture in the chapter. We show that with the proposed architecture, spectrum programmability can be decoupled from PHY layer processing and delegated to a virtualization layer, which is added between a set of virtual PHYs (vPHYs) and the HW/physical RF front-end. Additionally, we present an open-source and free proof of concept prototype developed to assess the performance of the proposed spectrum hypervisor architecture in real-world experiments.

Then, in Chapter 4, we propose an open-source SDR-based framework for intelligent spectrum sharing research in the context of next-generation wireless networks. The proposed framework can be used to prototype and experimentally test disruptive intelligent algorithms and techniques devised to optimize the suboptimal use of today's radio spectrum dynamically. The motivation behind the design of such framework is the scarcity of spectrum, which is mainly caused by splitting it into fixed and exclusively licensed bands that are assigned over large and geographically defined regions. Therefore, the proposed framework was thought of to reap the total capacity out of all types of available spectrum bands. Initially, several related works are compared with and distinguished from our work. We show that the proposed framework presents several advantages and characteristics that are not found in those related works. The proposed framework is composed of three main modules (i.e., PHY layer, RF monitor, and Listen Before Talking (LBT)). Implementation aspects of these modules are described in detail. The PHY layer employs along with discontinuous transmission a new type of waveform called Filtered-OFDM (f-OFDM), which is more suited for coexistence with other nodes. The RF monitor module is proposed for conferring spectrum sensing abilities to the framework. We propose a three-stage algorithm based on a Cell-Average Constant False Alarm Rate (CA-CFAR) strategy to assess spectrum occupancy. The SW-based LBT module is implemented as a contention-based mechanism for medium access. Additionally, three use cases employing the proposed framework for prediction of transmission patterns, intelligent LTE-

Licensed Assisted Access (LTE-LAA) and Wireless Fidelity (Wi-Fi) Coexistence, and coordinated spectrum sharing schemes are presented and discussed in detail. These use cases show how the framework can dynamically optimize the use of radio spectrum, combating spectrum scarcity and leveraging intelligent spectrum sharing research for next-generation wireless networks. Finally, we present the results of several experiments attesting the performance and effectiveness of the proposed framework.

In Chapter 5, we introduce the SCATTER PHY layer, which is an enhancement/extension of the framework presented in Section 4. Different from 4, SCATTER PHY presents several enhancements on top of the existing framework. Compared to the framework presented in Section 4, SCATTER PHY is an open-source, highly optimized, and real-time configurable SDR-based physical layer for the research and development of novel intelligent spectrum sharing schemes and algorithms. It presents two concurrent bursty f-OFDM-based PHY layers that operate independently of each other in full or half-duplex modes. We propose a novel frame structure focusing on higher throughput and with control signals being sent by the robust and more reliable Maximum length sequences (M-sequences) instead of LTE-like control channels. SCATTER PHY employs FPGA-based Finite Impulse Response (FIR) filters that improve/speed up the PHY processing performance while mitigating Out-Of-Band Emissions (OOBE). Consequently, improved coexistence with other nodes is achieved. We propose a two-stage synchronization sequence detection algorithm, which improves the detection performance compared to an LTE-like approach. Additionally, we also propose efficient and accurate schemes for cyclic prefix (CP)-based Carrier Frequency Offset (CFO) estimation and fine-grained complex exponential correction signal. We implement a simple receive combining scheme where signals coming from the two independent PHYs are combined for improved performance in low Signal to Noise Ratio (SNR) scenarios.

1.5 Publications

The research results obtained during this PhD research have been published in scientific journals and presented at a series of international conferences. The following list provides an overview of the publications during my PhD research.

1.5.1 Publications in international journals (listed in the Science Citation Index [2])

1. **Felipe A. P. de Figueiredo**, Fabiano Mathilde, Dick Carrillo, and Ingrid Moerman. *A Framework for the Automation of LTE Physical Layer Tests.* Published in the Wireless Personal Communications Journal, 102, 293–307 (2018).

2. **Felipe A. P. de Figueiredo**, Xianjun Jiao, Wei Liu, and Ingrid Moerman. *Radio hardware virtualization for HW-defined wireless networks.* Published in the Wireless Personal Communications Journal, 100, 113–126 (2018).

3. **Felipe A. P. de Figueiredo**, Xianjun Jiao, Wei Liu, Ruben Mennes, Irfan Jabandzic, and Ingrid Moerman. *A spectrum sharing framework for intelligent next a generation wireless networks.* Published in the IEEE Access Journal, vol. 6, pp. 60704-60735, 2018.

4. **Felipe A. P. de Figueiredo**, Dragoslav Stojadinovic, Prasanthi Maddala, Ruben Mennes, Irfan Jabandzic, Xianjun Jiao, and Ingrid Moerman. *SCATTER PHY: An Open Source Physical Layer for the DARPA Spectrum Collaboration Challenge.* Published in the MDPI Electronics Journal, vol. 8, no. 11, 2019.

5. **Felipe A. P. de Figueiredo**, Ruben Mennes, Irfan Jabandzic, Xianjun Jiao, and Ingrid Moerman. *A baseband wireless spectrum hypervisor for multiplexing concurrent OFDM signals.* Published in the MDPI Sensors Journal, vol. 20, no. 4, 2020.

6. Ruben Mennes, Maxim Claeys, **Felipe A. P. de Figueiredo**, Irfan Jabandzic, Ingrid Moerman, and Steven Latré. *Deep learning-based spectrum prediction collision avoidance for hybrid wireless environments.* Published in the IEEE Access Journal, vol. 7, pp. 45818-45830, 2019.

7. Ruben Mennes, **Felipe A. P. de Figueiredo**, and Steven Latré. *Multi-Agent Deep Learning for Multi-Channel Access in Slotted Wireless Networks.* Published in the IEEE Access Journal, vol. 8, pp. 95032-95045, 2020.

8. Irfan Jabandzic, **Felipe A. P. de Figueiredo**, Maxim Claeys, Ruben Mennes, Ingrid Moerman, and Spilios Giannoulis. *A Dynamic Distributed Multi-Channel TDMA Slot Management Protocol for Ad Hoc Networks.* Submitted to the IEEE Access Journal, 2021.

[2] The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index Expanded, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

1.5.2 Publications in book chapters

1. Xianjun Jiao, Ingrid Moerman, Wei Liu, and **Felipe A. P. de Figueiredo**, *“Radio Hardware Virtualization for Coping with Dynamic Heterogeneous Wireless Environments”*. Chapter in the book *Cognitive Radio Oriented Wireless Networks*, ISBN: 978-3-319-76206-7, Springer, 2018.

2. **Felipe A. P. de Figueiredo** and Fabbryccio A.C.M. Cardoso, *“An Efficient FPGA-based Frequency Shifter for LTE/LTE-A Systems”*. Chapter in the book *Field Programmable Gate Arrays (FPGAs) II*, ISBN: 978-1-83881-057-3, IntechOpen, 2020.

1.5.3 Publications in international conferences (listed in the Science Citation Index [3])

1. **Felipe A. P. de Figueiredo**, Fabiano Mathilde, Dick Carrillo, Luiz R Pizzini, and Ingrid Moerman. *Framework for automated tests of LTE physical layers.* Published in proceedings of the International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2017, Cusco, Peru, 2017.

2. **Felipe A. P. de Figueiredo**, Dragoslav Stojadinovic, Prasanthi Maddala, Ruben Mennes, Irfan Jabandzic , Xianjun Jiao, and Ingrid Moerman. *SCATTER PHY: a physical layer for the DARPA Spectrum Collaboration Challenge.* Published in proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySpAN), 2019, p.491-496, Newark, NJ, USA, 2019.

3. **Felipe A. P. de Figueiredo**, N.F.T. Aniceto, J. Seki, Ingrid Moerman,and G. Fraidenraich. *Comparing f-OFDM and OFDM performance for MIMO systems considering a 5G scenario.* Published in proceedings of the IEEE 2nd 5G World Forum (5GWF), 2019, p.491-496, Dresden, Germany, 2019.

4. Dragoslav Stojadinovic, **Felipe A. P. de Figueiredo**, Prasanthi Maddala, Ivan Seskar and Wade Trappe. *SC2 CIL : evaluating the spectrum voxel announcement benefits.* Published in proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySpAN), 2019, p.497-502, Newark, NJ, USA, 2019.

[3] The publications listed are recognized as ‘P1 publications’, according to the following definition used by Ghent University: P1 publications are proceedings listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

5. Spilios Giannoulis, Carlos Donato, Ruben Mennes, **Felipe A. P. de Figueiredo**, Irfan Jabandzic, Yorick De Bock, Miguel Camelo, Jakob Struye, Prasanthi Maddala, Michael Mehari, et al. *Dynamic and collaborative spectrum sharing: the SCATTER approach.* Published in proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySpAN), 2019, p.473-478, Newark, NJ, USA, 2019.

6. Miguel Camelo, Adnan Shahid, Jaron Fontaine, **Felipe A. P. de Figueiredo**, Eli De Poorter, Ingrid Moerman, and Steven Latré. *A semi-supervised learning approach towards automatic wireless technology recognition.* Published in proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySpAN), 2019, p.420-429, Newark, NJ, USA, 2019.

1.5.4 Publications in other international conferences

1. **Felipe A. P. de Figueiredo**, Wei Liu, Xianjun Jiao, and Ingrid Moerman. *Demo: packetized-LTE physical layer framework for coexistence experiments.* Published in proceedings of the ACM Conference on Embedded Network Sensor Systems (SenSys), 2017, p.1-2, Delft, The Netherlands, 2017.

2. Xianjun Jiao, Wei Liu, Muhammad Aslam, **Felipe A. P. de Figueiredo**, Ingrid Moerman, and Filip Louagie. *From laboratory to the field : an open source software defined Radio project coupled with native Linux driver framework.* Published in proceedings of the Wireless Innovation Forum European Conference (WinnComm), 2019, Berlin, Germany, 2019.

1.5.5 Patent applications

1. Xianjun Jiao, Wei Liu, **Felipe A. P. de Figueiredo**, and Ingrid Moerman. *Communication hardware virtualization.*. WO2019057731. IMEC VZW, Universiteit Gent. Priority date: 19 September 2017. Publication date: 28 March 2019.

References

[1] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020”, White paper, February 2016.

[2] Bakhshi, Taimur. ”State of the art and recent research advances in software defined networking.” Wireless Communications and Mobile Computing 2017 (2017).

[3] Hu, Honglin, Hsiao-Hwa Chen, Peter Mueller, Rose Qingyang Hu, and Yun Rui. ”Software defined wireless networks (SDWN): Part 1 [guest editorial].” IEEE Communications Magazine 53, no. 11 (2015): 108-109.

[4] Nick Feamster, Jennifer Rexford, and Ellen Zegura. 2014. The road to SDN: an intellectual history of programmable networks. SIGCOMM Comput. Commun. Rev. 44, 2 (April 2014), 87–98. DOI:https://doi.org/10.1145/2602204.2602219

[5] C. Liang and F. R. Yu, ”Wireless Network Virtualization: A Survey, Some Research Issues and Challenges,” in IEEE Communications Surveys Tutorials, vol. 17, no. 1, pp. 358-380, Firstquarter 2015, doi: 10.1109/COMST.2014.2352118.

[6] M. Richart, J. Baliosian, J. Serrat and J. Gorricho, ”Resource Slicing in Virtual Wireless Networks: A Survey,” in IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 462-476, Sept. 2016, doi: 10.1109/TNSM.2016.2597295.

[7] Chih-Lin, I., Huang, J., Duan, R., Cui, C., Jiang, J. and Li, L., 2014. Recent progress on C-RAN centralization and cloudification. IEEE Access, 2, pp.1030-1039.

[8] Hadzialic, Mesud, Branko Dosenovic, Merim Dzaferagic, and Jasmin Musovic. ”Cloud-RAN: Innovative radio access network architecture.” In Proceedings ELMAR-2013, pp. 115-120. IEEE, 2013.

[9] Mingjie He, Ying Yang and Weidong Wang, ”Distributed base station mobility strategy and load balancing in wireless networks,” 2013 International Conference on Wireless Communications and Signal Processing, Hangzhou, China, 2013, pp. 1-6, doi: 10.1109/WCSP.2013.6677057.

[10] Pompili, Dario, Abolfazl Hajisami, and Hariharasudhan Viswanathan. ”Dynamic provisioning and allocation in Cloud Radio Access Networks (C-RANs).” Ad Hoc Networks 30 (2015): 128-143.

[11] A. Checko, A. P. Avramova, M. S. Berger and H. L. Christiansen, ”Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings,” in Journal of Communications and Networks, vol. 18, no. 2, pp. 162-172, April 2016, doi: 10.1109/JCN.2016.000025.

[12] D. Wubben et al., ”Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN,” in IEEE Signal Processing Magazine, vol. 31, no. 6, pp. 35-44, Nov. 2014, doi: 10.1109/MSP.2014.2334952.

[13] C. Chang, R. Schiavi, N. Nikaein, T. Spyropoulos and C. Bonnet, ”Impact of packetization and functional split on C-RAN fronthaul performance,” 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016, pp. 1-7, doi: 10.1109/ICC.2016.7511579.

[14] Chih-Lin, I., Jinri Huang, Ran Duan, Chunfeng Cui, Jesse Jiang, and Lei Li. ”Recent progress on C-RAN centralization and cloudification.” IEEE Access 2 (2014): 1030-1039.

[15] P. Rost et al., ”Benefits and challenges of virtualization in 5G radio access networks,” in IEEE Communications Magazine, vol. 53, no. 12, pp. 75-82, Dec. 2015, doi: 10.1109/MCOM.2015.7355588.

[16] G. Staple and K. Werbach, ”The end of spectrum scarcity [spectrum allocation and utilization],” in IEEE Spectrum, vol. 41, no. 3, pp. 48-52, March 2004, doi: 10.1109/MSPEC.2004.1270548.

[17] Berlemann, Lars, and Stefan Mangold. Cognitive radio and dynamic spectrum access. J. Wiley Sons, 2009.

[18] B. D. Omkarappa, J. Benseny Quintana and H. Hämmainen, ”Study of spectrum scarcity and community wireless network operators in India,” 2017 Internet of Things Business Models, Users, and Networks, Copenhagen, 2017, pp. 1-5, doi: 10.1109/CTTE.2017.8260995.

[19] J. J. Popoola and R. van Olst, ”Application of neural network for sensing primary radio signals in a cognitive radio environment,” IEEE Africon ’11, Victoria Falls, Zambia, 2011, pp. 1-6, doi: 10.1109/AFRCON.2011.6072009.

[20] D. Cabric, S. M. Mishra and R. W. Brodersen, ”Implementation issues in spectrum sensing for cognitive radios,” Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004., 2004, pp. 772-776 Vol.1, doi: 10.1109/ACSSC.2004.1399240.

[21] Paul Tilghman, ”If DARPA Has Its Way, AI Will Rule the Wireless Spectrum”, IEEE Spectrum, May 2019. [Online at] https://spectrum.ieee.org/telecom/wireless/if-darpa-has-its-way-ai-will-rule-the-wireless-spectrum

[22] Chih-Lin I,; Jinri Huang,; Changming Bai,; Ran Duan,; Rongwei Ren, ”Wireless networks virtualization” (Telecommunications, 2017), ’5G Wireless Technologies’, Chap. 10, pp. 341-361, DOI: 10.1049/PBTE069E_ch10 IET Digital Library, https://digital-library.theiet.org/content/books/10.1049/pbte069e_ch10

[23] Paul, S.; Seshan, S. ”Technical Document on Wireless Virtualization”. GENI: Global Environment for Network Innovations, Technical Report, 2006.

[24] S. N. Khan, A. Kliks, Tao Chen, M. Mustonen, R. Riggio and L. Goratti, ”Virtualization of spectrum resources for 5G networks,” 2017 European Conference on Networks and Communications (EuCNC), Oulu, 2017, pp. 1-5, doi: 10.1109/EuCNC.2017.7980749.

[25] N. N. Sapavath and D. B. Rawat, ”Wireless Virtualization Architecture: Wireless Networking for Internet of Things,” in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5946-5953, July 2020, doi: 10.1109/JIOT.2019.2942542.

[26] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty. 2006. ”NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey”. Comput. Netw. 50, 13 (15 September 2006), 2127–2159. DOI:https://doi.org/10.1016/j.comnet.2006.05.001

[27] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. 2009. White space networking with wi-fi like connectivity. SIGCOMM Comput. Commun. Rev. 39, 4 (October 2009), 27–38. DOI:https://doi.org/10.1145/1594977.1592573

[28] M. M. Sohul, M. Yao, T. Yang and J. H. Reed, ”Spectrum access system for the citizen broadband radio service,” in IEEE Communications Magazine, vol. 53, no. 7, pp. 18-25, July 2015, doi: 10.1109/MCOM.2015.7158261.

[29] K. Tan, H. Shen, J. Zhang and Y. Zhang, ”Enable flexible spectrum access with spectrum virtualization,” 2012 IEEE International Symposium on Dynamic Spectrum Access Networks, Bellevue, WA, USA, 2012, pp. 47-58, doi: 10.1109/DYSPAN.2012.6478115.

[30] L. Cao, L. Yang and H. Zheng, ”The Impact of Frequency-Agility on Dynamic Spectrum Sharing,” 2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN), Singapore, 2010, pp. 1-12, doi: 10.1109/DYSPAN.2010.5457889.

[31] Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat. 2008. Learning to share: narrowband-friendly wideband networks. SIGCOMM Comput. Commun. Rev. 38, 4 (October 2008), 147–158. DOI:https://doi.org/10.1145/1402946.1402976

[32] Lei Yang, Wei Hou, Lili Cao, Ben Y. Zhao, and Haitao Zheng. 2010. Supporting demanding wireless applications with frequency-agile radios. In Proceedings of the 7th USENIX conference on Networked systems design and implementation (NSDI’10). USENIX Association, USA, 5.

[33] Ian F. Akyildiz, Pu Wang, Shih-Chun Lin, ”SoftAir: A software defined networking architecture for 5G wireless systems”, Computer Networks, Volume 85, 2015, Pages 1-18, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2015.05.007

[34] J. Liu, T. Zhao, S. Zhou, Y. Cheng and Z. Niu, ”CONCERT: a cloud-based architecture for next-generation cellular systems,” in IEEE Wireless Communications, vol. 21, no. 6, pp. 14-22, December 2014, doi: 10.1109/MWC.2014.7000967.

[35] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale and M. R. Souryal, ”Deep Learning Classification of 3.5-GHz Band Spectrograms With Applications to Spectrum Sensing,” in IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 2, pp. 224-236, June 2019, doi: 10.1109/TCCN.2019.2899871.

2

Radio Hardware Virtualization for Software-Defined Wireless Networks

Virtualization and softwarization are key enablers for accommodating the growing demand in data services related to 5G. Therefore, in this chapter, we propose a radio HW virtualization oriented transceiver architecture aimed at bringing the softwarization and virtualization techniques of wireless networks down to the PHY and MAC layers.

⋆ ⋆ ⋆

This chapter is based on the homonymous article by

Felipe A. P. de Figueiredo, Xianjun Jiao, Wei Liu, and Ingrid Moerman.

Published in Wireless Personal Communications, vol. 100, pp. 113–126, March 2018.

Abstract SDN is a promising architecture for the next-generation Internet. SDN can achieve Network Function Virtualization (NFV) much more efficiently than conventional architectures by splitting the data and control planes. Though SDN emerged first in wired networks, its wireless counterpart SDWN has been attracting an increasing amount of interest in recent years. Wireless networks have some distinct characteristics compared to the wired networks due to the wireless channel

dynamics. Therefore, wireless network controllers present some extra degrees of freedom, such as taking measurements against interference and noise or adapting channels according to the radio spectrum occupation. These specific characteristics bring about more challenges to wireless SDNs. Currently, SDWN implementations are mainly using customized firmware, such as OpenWRT [1], running on an embedded application processor in commercial Wi-Fi chips, and restricted to layers above lower MAC. This limitation comes from the fact that radio HW usually requires specific drivers, which have proprietary implementations by various chipset vendors. Hence, it is difficult, if not impossible, to achieve virtualization and softwarization of the radio HW in that case. However, this situation has been changing as SDR systems open up the entire radio communication stack to radio hobbyists and researchers. The bridge between SDR and SDN will make it possible to bring the softwarization and virtualization of wireless networks down to the PHY and lower-MAC layers, which will unlock the full potential of SDWN.

A very promising application of radio HW virtualization and softwarization is on the development of multi-wireless-standard multi-channel Access Points (APs). Radio HW virtualization and softwarization can be used to design radios to operate with diverse wireless standards, exhibiting low processing latency, fulfilling the most diverse requirements for traffic loads, and presenting the ability to operate in the same wireless environment with a common coordination agent, which leads to less interference and efficient spectrum usage. An interesting use case for this application is its deployment at the edge of C-RAN since such networks encounter challenges when low latency services and diverse access technologies are expected over non-fiber front-hauls. However, developing equipment for this kind of application is quite difficult as multiple radio chips and drivers have to be integrated and coordinated.

Therefore, this chapter investigates the necessity and feasibility of extending the virtualization and softwarization of wireless networks towards the radio HW. An SDR architecture is proposed for radio HW virtualization to facilitate SDWN design and experimentation. The proposed SDR architecture is developed on a single chip radio platform including HW accelerators wrapped by a SW Application Programming Interface (API), which offer the following capabilities: (1) concurrent data transmission over multiple virtual radios; (2) run-time composition and parametric control of the radios; and (3) radio resource slicing, supporting independent operation of multiple standards in different bands, time slots or beams. Such architecture offers a fast development cycle, as only SW programming is required for creating and manipulating multiple virtual radios. The architecture further achieves efficient utilization of HW resources, as HW accelerators can be shared by multiple virtual radios. Consequently, we do believe that by adopting the proposed virtualization-oriented HW accelerated design presented here, an all-layer end-to-end high-performance SDWN can be achieved.

2.1 Introduction

SDN is a promising concept at the networking level, it decouples the network control and data forwarding functions, allows directly programmable network control, and provides diverse network services to a variety of applications [2]. Before the introduction of the SDN concept, there was an increasing amount of labels/headers appended to packets, to support various kinds of protocols for different services on the Internet, which greatly increased the processing burden on the edge routers and switches. SDN solves the issue by using a disruptive design that separates data and control planes: routers and switches become dumb devices, which are only responsible for forwarding data according to the controller's instructions. The controller applies slow-varying configurations on the data forwarding devices, to slice/allocate the network resources to different types of services during runtime. Such an approach allows virtualizing a single physical network into multiple, independent and heterogeneous logical network domains, with each domain serving a specific category of traffic flow in the most appropriate way. The SDN approach is very encouraging, but has been basically designed for wired networks and mainly involves the higher layers of the protocol stack, e.g., layer 4 to 7 of the Open Systems Interconnection (OSI) model. It primarily provides transport capacity and service differentiation up to the edge router of wired networks.

Wireless networks benefiting from the flexibility offered by virtualization in SDN are known as SDWN, however, it must be emphasized that the wireless medium, i.e., the radio spectrum, has very different properties than the ones exhibited by wired networks. In wired networks, a port of a switch or router is always connected to optical fiber or Unshielded Twisted Pair (UTP) cables. Multiple ports are equivalent to multiple isolated non-interfering communication links with a constant data rate. As a result, Ethernet is ubiquitous in the wired network. On the other hand, the wireless medium is not isolated but shared. In wireless networks, there is interference when multiple links are simultaneously running in the same or adjacent radio spectrum bands. Besides, the data rate of a wireless link is dynamic, due to the variations in the distance (i.e., mobility), channel conditions (e.g., heavily shaded or Line-of-Sight), unpredicted interference from other co-located wireless technologies or radiating devices (e.g., microwave ovens). Hence, unlike wired networks that are dominated by Ethernet or optical links that have a deterministic capacity, wireless networks are non-deterministic and established upon many heterogeneous PHY and MAC layer standards, with each standard serving a different type of traffic flow. For example, LoRa [3] and SigFox [4] are used for long-range low rate sensor data collection, while Zigbee is designed for short-mid range low-mid rate sensor networks [5]; Bluetooth is known for short-range accessory communications [6]; Wi-Fi is devised for short-range high throughput applications [7]; 2nd Generation (2G)/3rd Generation (3G)/4th Generation (4G) mobile networks serve mid-high throughput terminals over a mid-long range [8, 9]; etc.

From an end-to-end communication point of view, the different traffic flows are characterized by different QoS requirements, such as, for example, latency and throughput. The main idea here is that, in wireless communications networks, one technology can hardly meet all requirements and can not give firm guarantees to QoS requirements. The lack of coordination and interaction among all the wireless network standards can jeopardize the overall performance of a network. The versatility in the wireless network's PHY and link layers is somewhat comparable to the era before SDN's appearance in the wired network, where various headers are appended into packets to support different services. Therefore, the logical next step in wireless networks is the achievement of runtime configuration across the diverse wireless standards by applying the SDN concept. This implies two requirements: (i) the lower layer radio stack needs to be more flexible to support runtime configuration and virtualization; (ii) the conventional SDN paradigm needs to be extended to counteract the uncertainties in wireless networks, by taking measurements to optimize the radio resource allocations (e.g., spectrum, time, and space).

One very interesting and encouraging application of virtualization and softwarization of the radio HW is in the development of solutions for coping with dynamic and heterogeneous wireless environments. In this application, virtualized and softwarized radio HW is used at the edge of C-RANs to independently serve diverse RATs with low latency over the same HW, i.e., the radio HW is shared among several wireless standards. Virtualization and softwarization techniques, can be easily applied to C-RAN, since all layers of RAN protocol, including the PHY layer, are implemented in SW running at cloud centers. By using virtualization and softwarization, it is no longer necessary to map each BBU SW instance to one physical server. Multiple BBU instances can share the same physical server, or multiple physical servers can serve as a supercomputer to run an ultra-high bandwidth BBU. This way, BBU instances can be created, destroyed or migrated according to dynamic requirements.

Operating in parallel with operators' RAN, there are many other types of wireless networks and standards, such as Wi-Fi [10] for internet or intranet access, 802.15.4/Zigbee [11] and 802.11ah [12] for short-to-middle range IoT applications, as well as LoRa [13] and SigFox [14] for long-range IoT, remain competitive. Besides, new applications are emerging, serving as a driving force of network technologies, which include robot or Unmanned Aerial Vehicle (UAV) control, vehicle-to-vehicle communications for autopiloting, Virtual Reality (VR), and real-time gaming. New standards or new features are needed over existing standards to interact with diverse physical worlds.

Furthermore, spectrum sharing is a common issue for many technologies mentioned above, especially for those operating in the ISM bands. In practice, the coordination among these technologies is either very hard to achieve or not present at all. Unlike the case of C-RAN, an operator can run multiple standards and base stations at one location – cloud center, the application here run in different physical AP/gateways, which are owned by different enterprises or even private home owners. This situation may lead to inefficient usage of spectrum, and severe QoS degradation for specific applications due to interference from heterogeneous technologies.

A piece of equipment supporting the parallel operation of multiple standards and multiple channels is a promising approach to deliver services with diverse QoS, in terms of optimized wireless access and spectrum utilization efficiency. For low latency applications, HW solutions, which could be ASIC or FPGA based, are more appropriate than SW-based ones. A problem is that one chip always is implemented for one or only a few standards, and operates at one frequency channel at a time. One option is to construct a piece of equipment with multiple chips to support multiple standards and channels, though this will be not only costly but also inconvenient to program and coordinate from the perspective of the developer — ASICs with dedicated configurations, drivers and inter-chip communication link become necessary prerequisites. For a multi-chip design, features commonly supported in C-RAN, such as dynamic computing resource sharing and migrating among cells/channels, are very hard to realize (if not impossible). This is because multi-chip HW design implies fixed HW resource allocation. An integrated new ASICs could be designed to merge multiple chips into a single one, but simply merging the chips won't give dynamic resource manipulation among intra ASICs blocks. Besides, the long design cycle and high development cost of ASICs would be a big obstacle on the road.

In this chapter, an architecture supporting low latency processing via HW (FPGA) accelerators, which is still flexible enough to support multi-standard multi-channel virtualization, is proposed to meet the requirement of diverse wireless access at the network edge. A demonstration is also implemented based on a SDR platform composed by a Xilinx Zynq SoC and an Analog Devices RF front-end.

In the remainder of this chapter, we first present a comprehensive view on the status of various efforts towards SDWN in Section 2.2. Next, in Section 2.3, an end-to-end view of the SDN-enabled wireless network from the ORCA project [15] is given. Then, Section 2.4 presents a requirement analysis and a novel architecture for radio HW virtualization to support the ORCA vision. In 2.5, we present an latency analysis followed by a demonstration and experimental validation in Section 2.6. Finally, we conclude this work in Section 2.7.

2.2 State of the art analysis

This section begins with the recent progress in the field of flexible and generic physical layer radio implementations, and the trend towards more dynamic spectrum allocation schemes; then we move on to two representative ways of real-life SDWN practices.

2.2.1 Evolution towards a flexible PHY

At the radio level, we have observed the emergence of SDR. An SDR is a radio communication system where transceiver components that are typically implemented on ASIC, e.g., digital mixers, filters, equalizers, modulators/demodulators, multiple antenna techniques, etc., are instead implemented on SW on a host computer or on an embedded system equipped with programmable HW like Application-Specific Instruction set Processor (ASIP) or FPGA. The concept behind SDR is very encouraging for the development of state-of-the-art PHY layer functionalities because SW programming allows faster development cycles. Therefore, many advanced and flexible physical layer techniques are available on SDR platforms, including Massive MIMO, full-duplex, mmWave, and various novel waveforms.

The main problem with SW implementation is the slower sequential execution of algorithms, even when multi-core or many-core Central Programming Units (CPUs) platforms or Graphics Processing Unitss (GPUs) are used, in contrast to a very fast execution and a very high degree of parallelization achieved with implementations on ASIC, ASIP or FPGA. For this reason, SDR development has so far mostly been limited to non-real-time physical layer development, as SW implementations do not always offer the fast execution times that are required for true networking experimentations, e.g., experiments requiring acknowledgment of MAC frames within a few microseconds. Recently, we have been observing limited yet increasing efforts to code more and more transceiver functionality on HW, e.g., FPGA, trading off SW flexibility for faster execution times, at the cost of higher design time. In both cases, SDR implementations are much more open, which gives the potential to support functionalities such as network virtualization on the lower communication stack.

2.2.2 Evolution towards generic PHY layer

Current wireless networks are composed of many different standards below the transport layer, which are necessary due to the versatility in the wireless medium and the traffic demands. In recent years, we have been noticing that individual physical layer standards are evolving towards each other to become more generic/homogeneous. For instance, NB-IoT has been developed as a subset of LTE to support low power and long-range IoT applications, which has similar capabilities as LoRA [16]. Besides, LTE also supports smaller Transmission Time Intervals (TTIs), which are complementary to the standard 1 ms TTI for serving low-latency applications [17]. Conventional Wi-Fi standards don't support flexible sub-carrier allocation, meaning that the spectrum resource cannot be sliced besides the selection of channels. The latest Wi-Fi standard 802.11ax is supposed to support the sub-carrier allocation feature, which is comparable to the resource blocks in LTE [18]. According to the standardization of 5G New Radio (NR), scalable OFDM numerology is proposed to support sub-carrier spacing ranging from 15 kHz (same as LTE) to 240 kHz (close to Wi-Fi 802.11a, 312.5 kHz), to enable the operation in a much wider radio spectrum and coverage areas [18, 19]. Without a doubt, the evolution towards a more generic/homogeneous physical layer in the wireless network will make the support of SDWN more convenient.

2.2.3 Evolution towards dynamic spectrum allocation

As stated previously, the radio communication link is not isolated by nature, but it could be achieved by enforcing the usage of a chunk of the spectrum only for a specific application, this is referred to as the licensed spectrum usage. This is the simplest approach, however, it is not efficient, since the static allocation causes waste of spectrum when there is no traffic demand from a given application. Some efforts are already being pushed to increase the utilization rate of the licensed spectrum, by allowing secondary usage of the spectrum without sacrificing the communication quality of the incumbents, e.g., TV white space [20] or spectrum sharing in radar bands [21]. The alternative to the simplest approach is the unlicensed spectrum access, where technologies share the medium with equal privileges. This approach is best represented by the current situation in ISM bands, where several technologies compete for spectrum access. However, the chances of over-the-air collisions increase when there is no coordination among devices/technologies, which is likely to trigger extended back off periods, leading to poor spectrum efficiency and lower QoS observed in the end-to-end communication links. The control and management functionalities in SDN could be borrowed to improve coordination between wireless network entities for spectrum usage. In addition to improving the efficiency of allocated radio spectrum, huge bandwidth can be harvested by extending wireless signal spectrum to mmWave band [22], such as 5G [18, 19] and 802.11ad [23].

Generally speaking, there is a trend going on for dynamic spectrum allocation, and many of the pioneer works in this area are based on SDR, focusing on its ability to rapidly adapt the operational parameters to achieve the optimal performance [48].

2.2.4 SDWN experiments on commercial Wi-Fi chipset

Some SDN experiments have already been carried out in the wireless network domain. An off-the-shelf Wi-Fi AP device can be split into multiple virtual APs on demand by flashing customized firmware, e.g., OpenWRT [25]. In this way, the bandwidth supported by the physical AP is sliced and can be allocated to different users or services. Seamless mobility among physical APs can be achieved by managing the virtual APs across multiple physical APs [26]. These functionalities are implemented in layers above the upper-MAC. It means that multiple virtual entities share a single physical layer and lower-MAC layer by Time-Division Multiplexing (TDM).

Although these efforts are important signs of progress towards SDWN, there are several significant limitations caused by the lack of SDN oriented physical and lower-MAC layers. As the virtual APs are created in a pure SW manner, the additional overhead caused by running multiple link services, e.g., authentication status maintenance, context switching, etc., upon a weak processor causes severe performance degradation in throughput [27, 28]. Furthermore, extra latency and jitter are introduced for a specific service/user, due to different virtual entities accessing the single physical link through time-division multiplexing, meaning that when one entity is accessing the physical link, the rest of the users/services are waiting for their turn.

2.2.5 SDWN experiments on SDR and cloud computing

Essentially, SDR aims to implement radio transceiver functionalities, which are traditionally realized in HW, in the SW domain. It is a promising candidate for physical layer implementation of SDWN, as demonstrated in use cases such as Cloud-based Radio Access Network or C-RAN [29]. In C-RAN systems, a Remote Radio Head (RRH) only performs conversion between the digitized baseband signal and the analog RF signal. The BBU is implemented in the servers on the cloud, which in turn performs all the necessary processing tasks of the physical layer. Empowered by the rather mature virtualization technologies in the computer science domain, the SW BBU can be created, allocated, migrated, and deleted on the fly. The BBU in the cloud might achieve comparable throughput as their HW counterparts, however, the performance in terms of latency is generally much worse. Fortunately, existing mobile network standards can tolerate a relatively large latency. For instance, LTE has a 1 ms TTI and 4 ms Hybrid Automatic Repeat Request (HARQ) feedback delay [9].

Although centralized BBU functions work well for some applications, it is difficult to serve applications with tight latency requirements. For example, self-driving functions relying on vehicle-to-vehicle communications require the base station to be located as close as possible to the vehicle in order to minimize the reaction time of the vehicle. For this type of use case, Mobile Edge Cloud (MEC) is a more suitable architecture than C-RAN. However, even MEC cannot softwarize the wireless standards with extremely low latency requirements, such as Wi-Fi. Wi-Fi's low MAC requires a node to acknowledge a successfully received packet within the duration of Short Inter Frame Space (SIFS) [7]. SIFS ranges from 16 μs down to 3 μs, depending on the specific variant of Wi-Fi standards (IEEE802.11a/b/g/n/ad, etc.). It is evident that this requirement cannot be met with the BBU entirely implemented in SW. The lower-MAC and HW coded physical layer need therefore to be tightly integrated in order to fulfill the necessary level of latency requirement.

2.3 ORCA's vision

The overall vision of the H2020 ORCA project is to drive end-to-end wireless network innovation by bridging real-time SDR and SDN. The project aims at exploiting the maximum flexibility at radio, medium access, and network levels, in order to meet a very diverse application requirements [30].

This vision is illustrated in Figure 2.1 and is further explained step-by-step using factory-of-the-future as the driving scenario. The manufacturing industry is one of the most demanding verticals with respect to ultra-low latencies, ultra-high reliability, ultra-high data rates, ultra-high availability, reliable indoor coverage in harsh environments (with a lot of metal structures) as well as energy-efficient and ultra-low communication costs for produced and connected goods. At the top of Figure 2.1 (beige color) different traffic classes can be observed corresponding to different application requirements. For the manufacturing scenario, a non-exhaustive list of Traffic Classess (TCs), can be identified. These TCs were inspired by [31, 32].

TC1: time-critical sensor/actuator control loop: bidirectional communication, low data rate (in the order of kbps), stringent timing requirements (below 1 ms cycle time, order 100 μs response time, below 1 μs jitter), ultra-high reliability (99.9999999 %), indoor, very short range (in the order of 10 m). Examples: motion control in printing machines, textile weaving machines, paper mills.

TC2: time-critical vision-controlled processes: bidirectional asymmetric communication ultra-high data rate (up to 10 Gbps), low latency (below 0.5 ms), high reliability (99.99999 %), indoor, short-range (10-100 m). Example: vision-controlled robot arms, vision-controlled quality inspection, wearables and augmented reality on the shop floor.

TC3: low-latency continuous medium-throughput: point-to-point and point-to-multipoint, moderate data rate (in the order of 10-100 kbps), low latency and jitter (both below 10 ms), ubiquitous coverage and high availability (indoor + on-site outdoor), mobility support, large autonomy. Example: voice communications

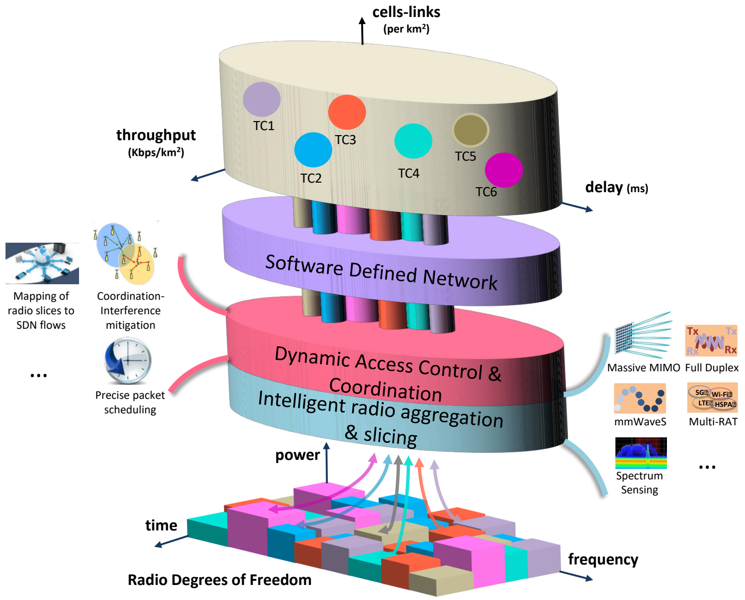


Figure 2.1: Network innovation driven by ORCA - the end-to-end view on SDN enabled wireless network [30].

between workers with headsets in the manufacturing hall.

TC4: correlated data capturing: moderate data rates (in the order of kbps up to 100 Mbps), moderate latency (10-100 ms), ultra-high time synchronization accuracy (below 100 ns), and moderate reliability (99.999 %), ubiquitous indoor coverage. Example: capturing of time-correlated sensor data on the shop floor to facilitate virtualized design processes that integrate simulator data with real-life data sensed during production.

TC5: non-time-critical in-factory communication: moderate data rates (in the order of kbps up to 100 Mbps), latency in the order of 100 ms (limited by human response times), moderate reliability (99.999 %) ubiquitous coverage and high availability (indoor + on-site outdoor), mobility support. Examples: interactions between humans and machines or robots, localization of assets and goods.

TC6: bursty traffic: non-time-critical (very large latencies allowed), large data volumes (in the order of MB up to 100 GB). Examples: sporadic SW/firmware updates of machines, temporary reconfiguration of machines.

TC7: best effort: low priority, no firm guarantees on data rates or latency, minimal shared capacity, ubiquitous coverage (indoor-outdoor). Example: typical Internet application (email, web surfing).

The current radio technologies lack capabilities with respect to wireless performance, management of heterogeneous devices, technology interoperability, and application (traffic) demands. Flexible and seamless connectivity across different Radio Access Technologies (RATs) will be required in order to instantaneously adapt the capacity and mobility needs to changing environments and application demands. A first approach to deal with such very diverse traffic demands would be the application of SDN techniques. Instead of employing one physical network infrastructure to deal with all the different traffic classes, applying complex traffic algorithms or QoS scheduling mechanisms, the network infrastructure can be virtualized into separate and independent network infrastructures, applying the most appropriate protocols and resource sharing mechanisms to deal with a specific traffic class. This approach is called network slicing or vertical slicing. This is illustrated by the vertical, colored pipes in Figure 2.1. Each pipe in the figure represents a single network slice, architected, and optimized for the specific requirements of the applications supported by its traffic class. For the manufacturing scenario described above, this results in 7 different pipes. The main focus of SDN today is on wired networks (Ethernet, optical transport networks) and on layer 3. ORCA offers a wireless SDN, by extending the current SDN vertical slicing capabilities with lower layer wireless capabilities.

To that end, the vertical pipes (corresponding to different traffic classes) need to be mapped onto the radio resource grid (bottom of Figure 2.1), hereby maximally exploiting the radio degrees of freedom like time, frequency, and space. It is important to note that the space dimension allows the reuse of spectrum and time resources through space division multiple access (not shown in Figure 2.1). The radio resource grid corresponds to the overall capacity of the radio infrastructure. Each block in the radio resource grid represents a chunk of radio resources consuming a certain part of the airtime, spectrum, and space (controlled by the power setting for omnidirectional antennas or by a directional beam in 3D MIMO case) with a certain PHY configuration (modulation and coding scheme) providing a certain dynamic capacity (in terms of data payload it can carry). This capacity is dynamic, as it changes over time due to changes in the wireless environment (requiring adaptations to the PHY). The mechanism of mapping vertical pipes to radio resource blocks is called radio slicing. It is responsible for the dynamic allocation of available resource blocks in the radio resource grid over the different traffic classes.

The focus of the ORCA project is on wireless functionalities that are needed to extend the current SDN concepts. ORCA has no intention to develop new network-level SDN paradigms, but will align with other SDN-oriented initiatives (based on heterogeneous and cooperative networks integrated through SDN/NFV techniques) to ensure that ORCA developments are compliant with common SDN mechanisms. Therefore, the focus of this chapter is to propose enhancements to the data plane functionalities of wireless networks once this is necessary to support more advanced SDN control functionalities in the future of SDWN.

2.4 Radio HW Virtualization

In this section, we present an analysis for a solution that meets several important requirements. Additionally, we also present and discuss an architecture proposed to meet all the listed requirements.

2.4.1 Requirement Analysis

To support the SDN functionalities and ORCA architecture, requirement analysis is carried out targeting runtime reconfigurable SDR physical and lower-MAC layers. More specifically, the ORCA SDR architecture aims to meet the following requirements:

1. **RF Resource Slicing**: RF resource slicing is used to slice wireless resources, such as spectrum, time, and beams, i.e., space beams pointing to specific directions. As a generalized module, it should not stick to any specific standard. A practical choice is to use it as the last stage of the digital processing chain, just before the Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC). The module multiplexes/demultiplexes IQ sample streams from/to physical layer transceivers. The transceivers could be physical entities or logical/virtual entities.

   To perform multiplexing/demultiplexing in real-time under control parameters, this module needs high processing throughput and precise timing control (in the case of time slicing). For instance, a 4 antenna Wi-Fi RF front-end generates 2.56 Gbps, i.e., a data rate of 20 Mega samples per second (Msps) (IQ samples) x 32-bit per sample (16-bit I, 16-bit Q) x 4 antennas. Therefore, in order to multiplex several Wi-Fi transceivers in the frequency domain, a link with a high data rate is required.

2. **Multi-channel Transceiver**: Multiple concurrent transceiver instances are necessary to utilize radio resources for multiple concurrent beams or frequency channels, in this way, multiple simultaneous services/standards are supported by separate radio slices. A multi-channel transceiver can be achieved by implementing multiple physical instances or creating multiple logical instances from single or fewer physical instances. In terms of HW/computing resources occupancy, the latter is better. When the same set of physical resources are shared by multiple logical instances, the HW context switching speed is essential to support multiple instances.

   The core part of the physical layer is the transceiver chain. In general, a receiver should include synchronization, CFO estimation/correction, channel estimation, equalization, OFDM demodulation, deframing, decoding, etc. On the other hand, a transmitter should include framing, coding, modulation, filtering, precoding, OFDM modulation, etc. All these base-band processing can be virtualized, but of course, time-efficient accelerators have to be implemented so that they can be time-shared.

For low latency standards or time-critical services, the transceiver should have low processing delay and should therefore be implemented in ASIC or FPGA as a group of HW accelerated modules, also known as HW accelerators. For relaxed latency standards or services, it could be either a SW or HW based implementation.

3. **Context Switching Support**: In the computer science domain, when multiple programs/virtual-machines share the same CPU, they sleep and wake up frequently and quickly, triggered by user input, network packet arrival, or other CPU-generated interruption. Before sleep, the CPU's state needs to be saved for the instance. This can be done by saving the CPU's internal registers into the memory. Before waking up, a restoring operation is performed to make sure that the execution is resumed correctly.

   Compared with CPU, the radio transceiver functionality is more complicated. There are lots of internal stages, First-In First-Outs (FIFOs), buffers, state machines inside the radio transceiver, therefore, context saving and restoring are challenging operations when one radio transceiver is supposed to be shared or switched quickly among multiple users/services.

   Therefore, besides traditional radio transceiver functionalities, the design should also support fast HW-level context saving and restoring. With this feature, a high-performance transceiver can be used to process multiple IQ streams in a fast switching TDM manner. Along with IQ buffers for each stream and transceiver consuming IQ samples much faster than IQ incoming into each buffer, a buffer overflow can be avoided for each IQ stream. The size of a IQ buffer depends on the HW accelerator's switching time granularity. When one accelerator is serving a channel, all the signals from the other channels have to be buffered (to prevent signal loss before the accelerator is able to process them). For example, if we use one accelerator to process incoming Wi-Fi's OFDM symbols, which are 4 us long, of two shared channels, each channel should buffer at least one OFDM symbol before the accelerator can process that channel. One OFDM symbol is 4 us * 20 Msps * 32 bits/sample * 2 antennas = 5120 bits long.

   Through this way, multiple concurrent logical transceivers can be created from a single transceiver to serve multiple traffic classes, and therefore, multiple end-to-end virtual slices can be implemented efficiently without implementing multiple physical transceivers.

4. **Resource Slicing Controller**: In this SDR-SDN context, there are two types of resources. The first type refers to the chunk of radio resources that are allocated to a single radio slice (such as beams, spectrum, and time) and can be used by a signal for transmitting and receiving. The second type of resource refers to the operation mode of the transceivers. This type of resource is used to deliver services/traffic within a transceiver's radio slice. We call the first type of resource 'RF resource', and the second resource 'transceiver resource'. To use resources smartly and efficiently under diverse and dynamic requirements, control SW is needed for the real-time management of resources. Although the control SW, in general, is not computationally intensive, controlling resources in precise timing is needed when a time slot is used in Time Division Multiple Access (TDMA) MAC, such as multi-frequency TDMA, multi-beam TDMA.

5. **SDN Agent**: At the AP or edge of the wireless network, a traditional SDN controller might not offer appropriate control functionality toward the AP or base station, because wireless equipment's capabilities are more complicated than "just forwarding". However, the traditional SDN controller does know the requirements of traffic classes or users. Therefore an SDN agent that incorporates wireless domain knowledge and that is capable of interpreting (more abstract) SDN requirements and mapping those into control strategies of radio domain resources is needed.

2.4.2 Architecture Design for Implementation

To meet the requirements mentioned in section 2.4.1 in a flexible way, Figure 2.2 depicts an architecture design for implementation. The proposed architecture supports both HW-like low latency performance and SW-like flexibility [33]. The platform is composed of a RF front-end and a digital baseband chip. The RF front-end can be any of the widely used devices, such as the FMCOMMS2 [34] or Universal Software Radio Peripheral (USRP) [35]. To make a highly efficient design, the digital baseband chip should include not only HW/FPGA for high-performance low-latency operation but also a processor system to support control and management functionalities of higher network layers.

HW/FPGA implementation or HW/FPGA accelerated SW is necessary to achieve real-time operation of certain wireless standards, such as Wi-Fi. Even for a mobile operator, in the era of beyond LTE, 5G is also considering much lower latencies (at sub-millisecond) than LTE. All these trends imply that pure centralized/cloud-based SW solutions might not be the silver bullet for all the envisioned scenarios/-cases. However, simply using pure ASIC/FPGA designs to meet performance/latency requirements will lose the SW-like flexibility, such as virtualization and effective coordination among channels and standards. Exploring an architecture,

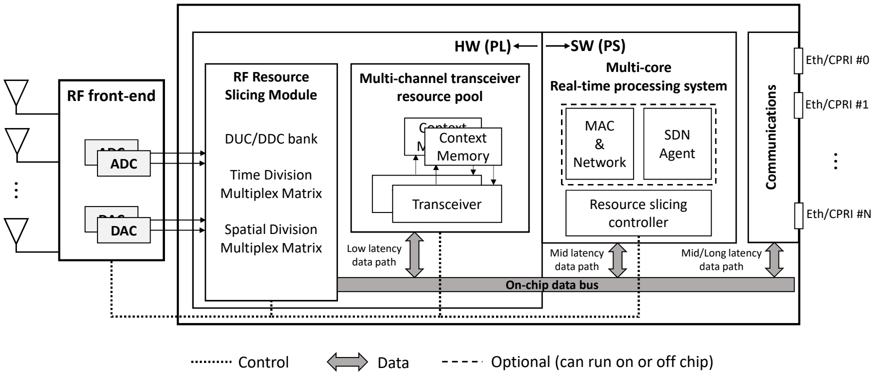

RF front-end
ADC
DAC
HW (PL)
SW (PS)
RF Resource Slicing Module
DUC/DDC bank
Time Division Multiplex Matrix
Spatial Division Multiplex Matrix
Multi-channel transceiver resource pool
Context Memory
Transceiver
Multi-core Real-time processing system
MAC & Network
SDN Agent
Resource slicing controller
Communications
Eth/CPRI #0
Eth/CPRI #1
Eth/CPRI #N
Low latency data path
Mid latency data path
Mid/Long latency data path
On-chip data bus
Control
Data
Optional (can run on or off chip)

Figure 2.2: Architecture proposed to meet the requirements presented in Section 4.

which has both HW/FPGA performance and SW flexibility, would be a more attractive option for next-generation wireless networks.

Therefore, SoC architectures are good candidates, such as the Xilinx Zynq SoC [37]. The Zynq SoC consists of two parts, the Programmable Logic (PL) part, which is a traditional FPGA, and the Processor System (PS) part, which includes an ARM-based multiprocessor system.

Two main modules are proposed to be implemented in HW/FPGA, namely, RF resource slicing module and multi-channel transceiver resource pool.

The first module is used to create the RF resource slices, such as antennas/beams, channels/bands, and time slots, which set the boundaries for transceiver operation in the second module. The RF resource slicing module is connected to the wideband multi-antenna RF frontend. It processes IQ samples to/from a specific slice under supervision of the resource slicing controller running on PS.

A pool of multi-channel transceivers is implemented in the second module with HW-level fast context switching support. Each transceiver is composed of several HW accelerated functions/components (e.g., digital mixers, filters, equalizers, modulators/demodulators, etc.). The transceivers construct data processing paths between diverse network traffic/service/user/standard and RF resources under control of the SW side. For the high-speed and low-latency on-chip connection between HW blocks and HW-SW, an Advanced Extensible Interface (AXI) stream bus can be used (the on-chip data bus in figure 2.2). The bandwidth requirement for this data bus is calculated as BW $=$ I/Q sampling rate$*$number of bits per sample$*$ number of antennas $*$ number of channels. For example, let's consider the case where we have 2 antennas, 2 Wi-Fi 20 MHz channels (20 Msps), and 32 bits per I/Q sample (16 bits I and 16 bits Q part), then the BW requirement is BW = 20 Msps * 32 bit/sample * 2 antennas * 2 channels = 2560 Mbps.

On-chip SW runs in the PS. Three main SW modules are needed: MAC and higher layer protocols, resource slicing controller, and SDN agent. As the proposed HW design supports virtualization, the corresponding MAC and higher layer protocols should also support creating corresponding multiple instances to handle diverse traffic streams in line with the SDN agent. To control the resources in real-time, the SW-based resource slicing controller communicates with the HW blocks through a AXI-Lite bus, which provides a register-like structure with reduced features and complexity [48]. As shown in Figure 2.2, the SW-based modules, MAC/higher layer protocols and SDN agent, can be run off chip depending on latency requirements, i.e., they can run at the pool of BBUs in the cloud center. The connection with the pool of BBUs is achieved though Common Public Radio Interface (CPRI) or Ethernet connections.

Slice specific IQ samples coming from the RF resource slicing module are routed via the on chip data bus to/from different destinations according to latency requirements. If the IQ samples belong to a low latency service (e.g., UAV and self-driving-car control) or standard Wi-Fi, their destination/source will be an on-chip multi-channel HW accelerated transceiver in the pool of transceivers. For IQ samples of radio slices that carry mid/long latency services, e.g., TV signals or standard LTE, they can be routed to pure SW domain: either on-chip PS or off-chip network interface (Ethernet, CPRI, etc.) until a processing unit in a host computer/cloud server is reached.

An issue that is addressed by the context memory block when the transceivers are shared among multiple slices is their context saving and restoring. In the proposed design, each transceiver has an associated context memory. When a transceiver is shared among multiple slices in TDM mode, the context memory block maintains context internally for each slice the associated transceiver serves. This is necessary to resume the transceiver's execution for a specific slice from its previous state before the interruption. If the HW accelerator processing is not finished before it has to switch to another channel, its intermediate states/signals are stored in the context memory block in case it switches back in the future to continue the processing of that channel by resuming the "paused" processing based on those stored intermediate states/signals. The context memory block can store the internal states of filters, correlators, FFTs/IFFTs, etc.

Needless to say, the SW design used to schedule diverse transceivers for different slices is also a challenging task on its own. This SW, which is implemented by the resource slicing controller, is essential to achieve effective spectrum utilization with the coordination of multiple standards/channels.

2.5 Latency Analysis

Latency is one of the key factors for different types of standards and applications. The system architecture of wireless technology is influenced by the latency re-

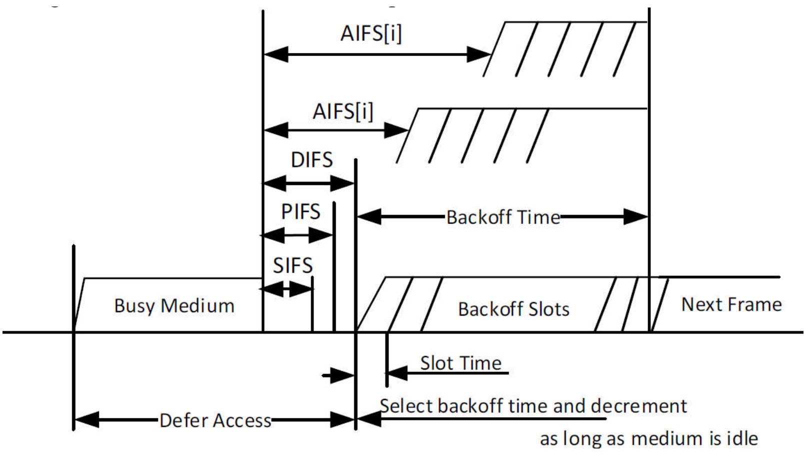


Figure 2.3: 802.11 interframe space relationships [10].

quirements and implementation feasibility of specific latency targets. In this section, we first investigate the latency requirements of mainstream wireless standards and then present the latency measurements of two types of SDR platforms, namely the USRP, and Xilinx Zynq SoC-based SDR.

2.5.1 Latency Requirement of Wi-Fi and LTE

According to Wi-Fi standard [10], the most critical latency requirement is from SIFS. The concept of interframe space is shown in Figure 2.3, which is retrieved from “Figure 10-4—Some IFS relationships” of 802.11-2016 standard [10].

As explained in the standard [10]: ”The SIFS is the time from the end of the last symbol, or signal extension if present, of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen on the WM. The SIFS shall be used before transmission of an ACK frame ...”. The SIFS specified in the standard is 10μs or 16μs, depending on band (2.4 GHz or 5 GHz) and PHY type (DSSS, OFDM, etc.). For the 60 GHz band system, which has directional multigigabit ability, SIFS is only 3us.

Unlike Wi-Fi’s CSMA/CA MAC mechanism, in LTE system everything is scheduled in advance, including acknowledgment of Hybrid Automatic Repeat Request (HARQ) process. Figure 2.4 shows LTE uplink HARQ procedure (reproduction of Figure 10.14 of [9]).

In uplink HARQ, the base station has 3 ms to acknowledge using the downlink Physical Hybrid ARQ Indicator Channel (PHICH) after it receives uplink IQ samples of Physical Uplink Shared Channel (PUSCH). According to the C-RAN white paper [29], the base station needs $800 \sim 900\mu$s after receiving IQ samples

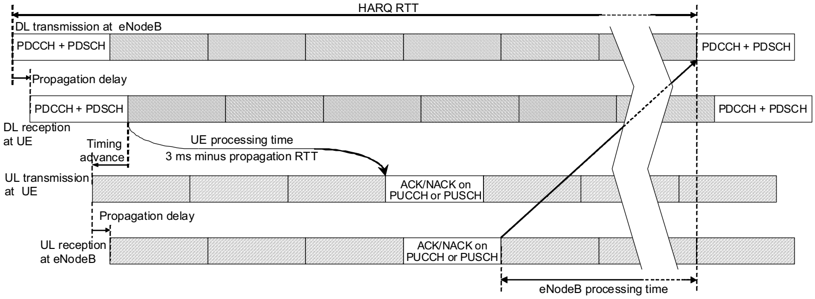


Figure 2.4: Timing diagram of the uplink HARQ (SAW - Stop And Wait) protocol [9].

of each 1 ms. So the round trip latency budget between RRH and BBU is around $100 \sim 200\mu$s.

According to CPRI over fiber specification [38], the maximum one-way latency of the CPRI link is 5μs. Besides CPRI latency, other latency overheads between RRH and BBU include physical distance, buffers, switches, cloud computer interface (Ethernet/PCIe/Operating System), etc. All together it should be less than the round trip latency budget of C-RAN fronthaul. CPRI over Ethernet [39] is a hot topic in the C-RAN fronthaul area, because not every place/area has optical fiber coverage.

2.5.2 Latency Measurement Results of USRP + Host Computer

USRP [35] SDR platform is the most widely used platform in the research community. In most cases, it is used jointly with the host computer, which can be (to some extent) regarded as a minimum version of a C-RAN server. The latency test is provided by the USRP Hardware Driver (UHD) [40] native example: $latency\_test$ [41]. Table 2.1 shows the measurement results of different USRPs combined with different computer communication links. Each latency result is measured and averaged over 10 round tests of 5, 10, and 25 Mega sampling rates.

Table 2.1: Round trip latency between RF frontend and host computer SW.

USRP type	Link type	Latency (μs)	Host computer configuration
X310	PCIe	79	Intel i7-6700 3.4 GHz, NI PCIe x4 card
X310	10 Gbps Ethernet	93	Intel i7-6700 3.4 GHz, Qlogic 57810 Eth
X310	1 Gbps Ethernet	101	Intel i7-6700 3.4 GHz, Intel i219-v Eth
B210/200mini	USB 3.0	66	Intel i7-6700 3.4 GHz, Intel controller
N210	1 Gbps Ethernet	103	Intel i7-6700 3.4 GHz, Intel i219-v Eth

The latency measurement results show that the host computer-based SDR architecture can't meet the SIFS requirement of the Wi-Fi system, but it is good enough to be used for a LTE system because the latency of $\sim 100\mu$s only consumes

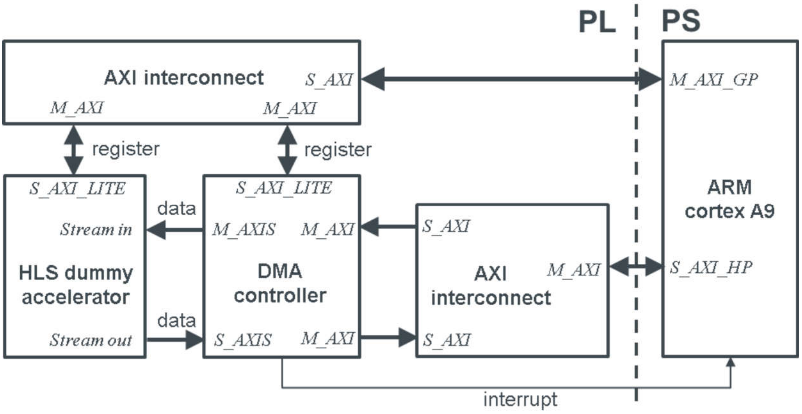


Figure 2.5: Diagram for latency test of the Zynq-7000 Platform.

a tiny portion of LTE processing time needed by HARQ processing. That is why there are already several host-computer based LTE systems, such as srsLTE [42], Open Air Interface (OAI) [43], and amarisoft [44], which work quite well. On the contrary, almost all Wi-Fi SDR implementations, such as NI 802.11 Application Framework [45] and Wireless Open-Access Research Platform (WARP) [46], use FPGA to do processing instead of the host computer.

2.5.3 Latency Measurement Results of Xilinx Zynq-7000 SoC

Xilinx Zynq-7000 All Programmable SoC [36] includes two parts: PL and PS. PL actually is a traditional FPGA, which can do the computation-intensive operation and latency critical tasks. PS is a multi-cores ARM cortex application processor, which is suitable to run control programs, higher layer protocols, and operating system. PL and PS are connected by multiple AXI high-speed buses, which have low latency and high throughput performance. We evaluate its latency using a dummy FPGA block interacting with an ARM testing program in bare metal mode.

The testing system was constructed as in Figure 2.5. The dummy FPGA acceleration block was designed by Xilinx HLS tool. The block receives data from PS in a streaming manner via Xilinx Direct Memory Access (DMA) module, and stream the processed result (same amount of data as input) back to PS via DMA. It takes 654 clock cycles for this FPGA block to process 128 input samples, each sample is represented as a 32-bit word on the 32-bit AXI stream bus. The configuration interface between PS, PL, and DMA controller is AXI_LITE, which is connected to the M_AXI_GP port of ARM. Datalink is an AXI stream, which is connected to the S_AXI_HP port of ARM. A DMA controller was used to convert AXI Memory Mapped interface (needed by PS) to the AXI Stream interface

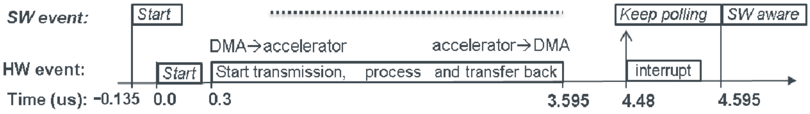

SW event:
Start
Keep polling
SW aware
DMA→accelerator
accelerator→DMA
HW event:
Start
Start transmission, process and transfer back
interrupt
Time (us): -0.135 0.0 0.3 3.595 4.48 4.595

Figure 2.6: Latency test result of Zynq-7000 SoC Platform.

(needed by streaming mode FPGA accelerator).

ZC706 Evaluation Board [37] for the Zynq-7000 XC7Z045 SoC is used to do the evaluation. Clock speeds of our design are as follows: AXI buses and PL run at 200 MHz, and ARM cortex-A9 processor running at 800 MHz. Xilinx Integrated Logic Analyzer (ILA) is inserted into the design for event recording with 5 ns resolution. The ARM SW event is recorded by writing a special value to the PL register and detecting this value via ILA. The latency profiling result is shown in Figure 2.6.

Figure 2.6 shows the event log of the entire round trip delay test captured by Xilinx ILA. First, the SW starts DMA transmission at -0.135μs; then the DMA controller receives the instruction at 0μs via AXI LITE register interface. After some internal preparation and buffering operation, the actual DMA transmission on the AXIS bus starts at 0.3μs; After 3.295μs, which is caused by the accelerator processing latency of 654 clocks at 200 MHz, the accelerator completes the data transfer of processing results back to DMA controller in streaming manner. Then DMA controller raises interrupt to PS at 4.48μs. Finally, the SW becomes aware of this event at 4.595μs. So the round trip latency between the FPGA accelerator and SW in PS is $4.595 + 0.135 - (3.595 - 0.3) = 1.435\mu$s. Note that this is a superior performance comparing to the latency of USRP variants, and it is even not a significant overhead compared to the Wi-Fi SIFS requirement (16μs or 10μs).

The reason for this higher performance is that Zynq 7000 is a System-on-Chip (SOC), where the FPGA (i.e., the PL part) is on the same chip as the ARM processor (i.e., the PS part). They are physically right beside each other. In that case, the piece of SW measuring the RTT was running on the ARM processor. This piece of SW is running on bare metal (i.e., no operating system involved), it is directly connected to the HW. On the other hand, the RTTs measured for the USRP+Host PC case were measured by an application running on user space (i.e., a piece of SW running on top of the Linux kernel).

2.6 Demonstration

A proof of concept demonstration of radio HW virtualization [33] has been developed based on a SDR platform composed of a FMCOMMS2 RF front-end and Xilinx ZC706 [37] evaluation board as shown in Figure 2.7.

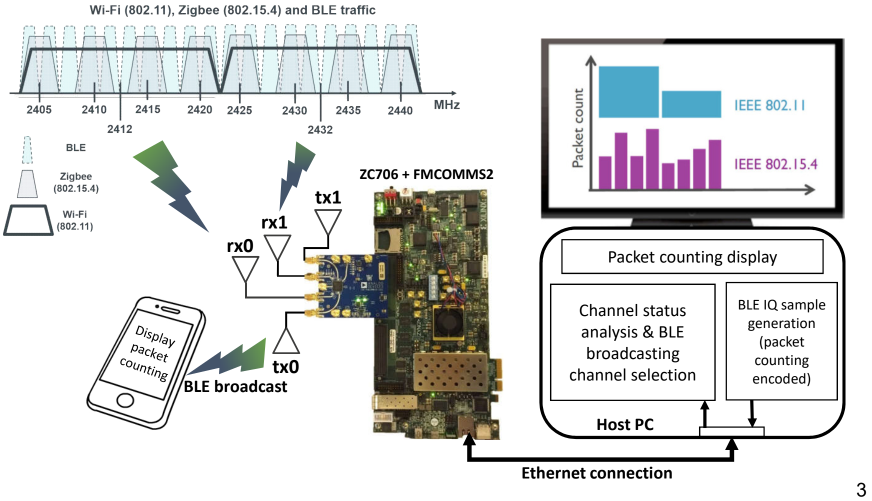


Figure 2.7: Demonstration of multiple virtual radios on a single chip.

The RF front end has 2 receiving antennas and 2 transmitting ones (Rx0, Rx1, Tx0, and Tx1 as shown in Figure 2.7), and works in 40 MHz bandwidth mode. A host computer is connected to the SDR platform for visualization and processing mid/long latency tasks. Three standards are covered in the demonstration, namely, Wi-Fi/802.11 20 MHz mode, Zigbee/802.15.4 2.4 GHz version, and Bluetooth Low Energy (BLE). In this demonstration, a Digital Down-converter (DDC) bank is implemented for the RF spectral resource slicing part. It slices the 40 MHz spectrum (partial 2.4 GHz ISM band) into two adjacent 20 MHz Wi-Fi channels, overlapping with eight 5 MHz ZigBee channels [47]. For each Rx antenna, the resource slicing module creates one 20 MHz Wi-Fi channel and four 5 MHz Zigbee channels (overlapped with Wi-Fi) with five DDCs. The central frequency of each Rx antenna's DDC can be tuned independently within the range of 40 MHz bandwidth. In the demonstration, Rx0 is tuned to cover the upper 20 MHz of 40 MHz bandwidth and Rx1 is tuned to cover the lower 20 MHz of 40 MHz bandwidth. A 9.6 Gbps AXI bus is used to construct the on-chip data bus. Dedicated AXI links are setup between RF resource slicing module, the pool of transceivers implementing the preamble detector and PS.

A dual-standard preamble detector (part of the baseband receiver), with fast HW context-maintenance support, is implemented for the transceiver resource part. Based on the FPGA design, the resource slicing controller SW running in the PS creates 10 virtual preamble detector instances out of the single FPGA preamble detector block to serve 10 input IQ sample streams (2 Wi-Fi, 8 ZigBee). From the user point of view, it is the same as having 10 parallel preamble detectors running concurrently in full-time, which can show packet count statistics of 10 concurrent live traffics in the air.

In addition, in order to make the demonstration more user friendly, a BLE transmitter is implemented in the host computer. In this demonstration, the BLE broadcasting is an example of mid/long latency service that is handled by the host computer or central/cloud server instead of the HW, i.e., the FPGA. The PS reports Wi-Fi and Zigbee packet counting results of each slice to the host computer via Ethernet. A program running in the host computer analyzes the channel status according to the packet counting report, then it: 1) selects the less busy BLE broadcasting channel (based on the packet count detected by the 10 virtual preamble detector instances) from three options: 2402 MHz, 2426 MHz, or 2480 MHz; 2) generates IQ samples for BLE broadcasting packet [49] with Wi-Fi and Zigbee packet count statistics information encoded; 3) sends BLE IQ samples to the PS in the SDR platform via Ethernet. The RF resource slicing module broadcasts the BLE packet into the air via the proper Tx slice (frequency, timing and antenna) under PS control. Finally, a user can read the BLE message, which contains Wi-Fi and Zigbee packet counting information, on their device's (mobile-phone, tablet, computer, etc.) screen with any general-purpose BLE scanner/sniffer application, for instance, LightBlue and BLE Scanner, which are freely available on Google's play store.

A dual-standard preamble detector (i.e., a 16-sample auto-correlation algorithm) is designed based on the similar structure of Wi-Fi and Zigbee's preambles. According to the respective standards, at Wi-Fi's baseband rate of 20 Msps and Zigbee's half baseband rate of 1 Msps, both preambles are created by repeating a 16-sample long random signal. The only difference between them is that Wi-Fi's preamble has 160 samples, which are generated by repeating the 16-sample signal 10 times and Zigbee repeats it 8 times, which results in a 128-sample long preamble.

C language is used to develop both FPGA and PS SW. By using Xilinx HLS C, the RF resource slicing and preamble detector, which is part of a transceiver, modules achieve 200 MHz clock speed. The AXI data bus runs at 64-bit 200 MHz mode. Each AXI 64-bit word contains two IQ samples (16-bit In-phase and 16-bit Quadrature sample for each antenna). In order to process 512 samples, which means 25.6 μs under Wi-Fi's baseband rate and 512 μs under half Zigbee's baseband rate, the preamble detector needs only 1094 clocks, i.e., 5.47 μs, according to the FPGA synthesis report. Therefore, it is fast enough to handle two Wi-Fi and eight Zigbee slices. This is due to the fact that the speed of the transceiver (i.e., preamble detector) consuming the IQ samples is higher than the speed of the ten Rx slices generating the IQ samples. The whole processing system won't lose any IQ-sample, since it works as if ten preamble detectors are running fully in parallel (i.e., logically in parallel). Differently from the technology proposed in [25], where multiple virtual APs share one Wi-Fi channel by time division, our

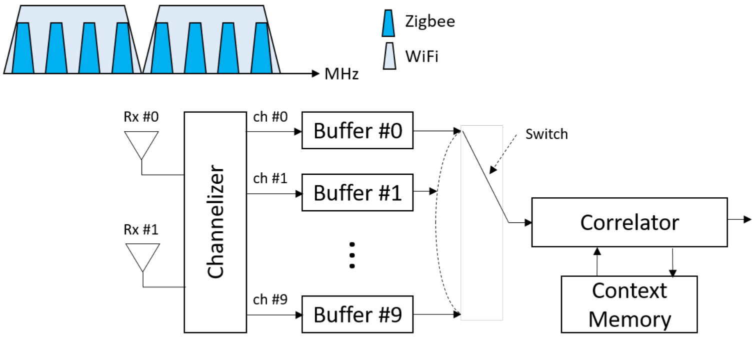


Figure 2.8: Architecture of the preamble detector.

implementation allows running multiple APs over multiple channels concurrently without the necessity of implementing standalone FPGA blocks for each channel.

For the control SW, a basic control slot length is 25.6 μs long (512 Wi-Fi's samples), which is also the basic time-slot length of resource slicing. Then, 20 slots compose a control period. Different tasks, such as the detection of ten preambles, BLE broadcasting, host computer communication, Tx/Rx frequency tuning, are scheduled to different time slots according to the upstream producing rate and latency requirement, just like a computer running multiple processes. By carefully arranging the tasks' schedules, there are then ten Rx instances and one Tx instance running fully in parallel from the end-user's point of view. In this way, a set of radio HW resources (RF front end, transceivers, etc.) can be virtualized into multiple instances.

Figure 2.8 illustrates how dual-standard preamble detection is achieved. As the preamble detector processes IQ samples faster than the 10 channels can produce samples, it can easily change context and process all the channel buffers without losing samples. The algorithm behind preamble detection is not new. The algorithm detects the multiple repetitions of the 16-sample random sequence using equation (2.1).

$$\text{corr[i]} = \frac{\| \sum_{i=0}^{N} S[i] * S^*[i-16] \|}{\sum_{i=0}^{N} S[i] * S^*[i]}, \tag{2.1}$$

where S and S^* are the received signal and its conjugate, and $N = 16$

What is new in the proposed design is the realization that WiFi and Zigbee preambles are created by repeating the same 16-sample structure. Based on that, we created a single piece of HW that detects both technologies by simply processing them faster than they are received.

In our implementation, we used the same RF front-end for both technologies. As far as we understand, the virtualization won't impact the specification of the front-end. For example, a Wi-Fi capable front-end would suffice to virtualize both technologies. We just need a front-end covering the wider bandwidth and probably some features such as low noise figure, high linearity, etc., meeting the requirements of the most stringent technology. Therefore, if one has a broadband front-end meeting the requirements of the technologies, then, it should used.

2.7 Conclusions

In this chapter, a radio HW virtualization oriented transceiver architecture is proposed to bridge the gap between the diverse real-world applications and the scarce RF resources. This architecture softwarizes the lowest part of the wireless network stack such as PHY and low-MAC layers while maintaining equally high performance and low latency as in the conventional HW-defined network. With this radio HW virtualization feature, the control plane can make efficient RF and HW resource utilization according to dynamic network traffic/service/standard requirements.

The proposed architecture allows pushing the virtualization present in cloud or host-computer domain to radio HW level to solve the spectrum sharing issue under diverse wireless access scenarios. By building a platform with a SW-HW co-design philosophy, researchers and developers can use it to create multiple concurrent virtualized instances from low-latency and high-performance HW/FPGA transceivers. With the proposed architecture, multiple wireless accesses can fully run in parallel, just like running multiple programs on the same CPU, to maximize utilization of FPGA transceivers. Additionally, HW accelerators can be shared between standards (if they have similar processing steps, like WiFi's and Zigbee's preamble detection) or channels.

The proof-of-concept demonstration shows the feasibility of the radio HW virtualization with limited HW resources. As the next step in the ORCA project, we will bridge real-time SDR and SDN with the help of radio HW virtualization and exploit maximum flexibility at PHY, MAC, and network levels, as a way to support very diverse application requirements by efficiently sharing the limited RF and transceiver resources.

Acknowledgment

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732174 (ORCA project).

References

[1] OpenWrt, ”OpenWrt Project”, [available at] https://openwrt.org/, retrieved on the 30th of September 2017.

[2] Singh, Sanjeev, and Rakesh Kumar Jha. ”A survey on software defined networking: Architecture for next generation network”, Journal of Network and Systems Management 25, no. 2, pp. 321-374, 2017.

[3] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang, ”A survey on LPWA technology: LoRa and NB-IoT”, ICT Express, vol. 3, no. 1, pp. 14-21, Mar. 2017.

[4] Mads Lauridsen, Benny Vejlgaard, Istvan Z. Kovacs, Huan Nguyen, and Preben Mogensen, ”Interference Measurements in the European 868 MHz ISM Band with Focus on LoRa and SigFox”, IEEE Wireless Communications and Networking Conference (WCNC), Mar. 2017.

[5] Ata Elahi, and Adam Gschwender, ”ZigBee Wireless Sensor and Control Network”, Prentice Hall Press, ISBN: 0137134851, 2009.

[6] Martin Sauter, ”From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband”, John Wiley & Sons, ISBN: 9781119346869, Aug. 2017

[7] IEEE Standards Association, ”Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, Sept. 1999.

[8] Ajay R. Mishra, ”Advanced Cellular Network Planning and Optimisation: 2G/2.5G/3G...Evolution to 4G”, John Wiley & Sons, ISBN: 9780470014714, Nov. 2006.

[9] Stefania Sesia, Issam Toufik, and Matthew Baker, ”LTE - The UMTS Long Term Evolution: From Theory to Practice”, John Wiley & Sons, ISBN: 9780470660256, Jul. 2011.

[10] ”IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band,” in IEEE Std 802.11b-1999 , vol., no., pp.1-96, 20 Jan. 2000, doi: 10.1109/IEEESTD.2000.90914.

[11] ”IEEE Standard for Low-Rate Wireless Networks,” in IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011) , vol., no., pp.1-709, 22 April 2016, doi: 10.1109/IEEESTD.2016.7460875.

[12] M. Taneja, ”802.11ah - LPWA interworking,” 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, 2016, pp. 441-446, doi: 10.1109/NETSOFT.2016.7502482.

[13] Martin Bor, John Vidler, and Utz Roedig, ”LoRa for the Internet of Things”, In Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN ’16). Junction Publishing, USA, 361–366, 2016.

[14] A. Lavric, A. I. Petrariu and V. Popa, ”Long Range SigFox Communication Protocol Scalability Analysis Under Large-Scale, High-Density Conditions,” in IEEE Access, vol. 7, pp. 35816-35825, 2019, doi: 10.1109/ACCESS.2019.2903157.

[15] ORCA Project, ”ORCA Orchestration and Reconfiguration Control Architecture”, [available at] https://www.orca-project.eu/, retrieved on the 25th of September 2017.

[16] Y.-P. Eric Wang, Xingqin Lin, Ansuman Adhikary, Asbjorn Grovlen, Yutao Sui, Yufei Blankenship, Johan Bergman, and Hazhir S. Razaghi, ”A Primer on 3GPP Narrowband Internet of Things (NB-IoT)”, IEEE Communications Magazine, vol. 55, no. 3, pp. 117-123, Mar. 2017.

[17] Kazuki Takeda, Li Hui Wang, and Satoshi Nagata, ”Industry Perspectives: Latency Reduction toward 5G”, IEEE Wireless Communications, vol. 24, no. 3, pp. 2-4, Jun. 2017.

[18] M. Shahwaiz Afaqui, Eduard Garcia-Villegas, and Elena Lopez-Aguilera, ”IEEE 802.11ax: Challenges and Requirements for Future High Efficiency Wi-Fi”, IEEE Wireless Communications, vol. 24, no. 3, pp. 130-137, Dec. 2016.

[19] Ali A. Zaidi, Robert Baldemair, Hugo Tullberg, Hakan Bjorkegren, Lars Sundstrom, Jonas Medbo, Caner Kilinc, and Icaro Da Silva, ”Waveform and Numerology to Support 5G Services and Requirements”, IEEE Communications Magazine, vol. 54, no. 11, pp. 90-98, Nov. 2016.

[20] Oliver Holland, Hanna Bogucka, and Arturas Medeisis, ”Opportunistic Spectrum Sharing and White Space Access: The Practical Reality”, John Wiley & Sons, ISBN: 978-1-118-89374-6, Jun. 2015.

[21] Francisco Paisana, Nicholas J. Kaminski, Nicola Marchetti, and Luiz A. DaSilva, ”Signal Processing for Temporal Spectrum Sharing in a Multi-radar Environment”, IEEE Trans. on Cognitive Communications and Networking, vol. 3, no. 2, pp. 123-137, Jun. 2017.

[22] Ming Xiao, Shahid Mumtaz, Yongming Huang, Linglong Dai, Yonghui Li, Michail Matthaiou, George K. Karagiannidis, Emil Björnson, Kai Yang, Chih Lin, and Amitava Ghosh, "Millimeter Wave Communications for Future Mobile Networks", IEEE Journal on Selected Areas in Communications, vol. 35, no. 9, pp. 1909-1935, Sept. 2017.

[23] Thomas Nitsche, Carlos Cordeiro, Adriana B. Flores, Edward W. Knightly, Eldad Perahia, and Joerg C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi [Invited Paper]", IEEE Communications Magazine, vol. 52, no. 12, pp. 132-141, Dec. 2014.

[24] Beibei Wang, and K.J. Ray Liu, "Advances in Cognitive Radio Networks: A Survey", IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 1, pp. 5-23, Feb. 2011.

[25] Bhanage, G., Vete, D., Seskar, I., and Raychaudhuri, D., "SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs", IEEE Global Telecommunications Conference (GLOBECOM), 2010.

[26] Peter Dely, Jonathan Vestin, Andreas Kassler, Nico Bayer, Hans Einsiedler, and Christoph Peylo, "CloudMAC - An OpenFlow based architecture for 802.11 MAC layer processing in the cloud", IEEE Globecom Workshops (GC Wkshps), Dec. 2012.

[27] Tausif Zahid, Fouad Yousuf Dar, Xiaojun Hei, and Wenqing Cheng, "A measurement study of a single-BSS software defined Wi-Fi testbed", IEEE International Conference on Computer Communication and the Internet (ICCCI), Oct. 2016.

[28] Tausif Zahid, Xiaojun Hei, and Wenqing Cheng, "Understanding the Design Space of a Software Defined Wi-Fi Network Testbed", International Conference on Frontiers of Information (FIT), Dec. 2016.

[29] Kulin Chen, and Run Duan, "C-RAN the road towards green RAN", China Mobile Research Institute, white paper, 2011.

[30] Tarik Kazaz, Wei Liu, Xianjun Jiao, Ingrid Moerman, Francisco Paisana, Clemens Felber, Vincent Kotzsch, Ivan Seskar, Tom Vermeulen, Sofie Pollin, Martin Danneberg, and Roberto Bomfin, "Orchestration and Reconfiguration Control", European Conference on Networks and Communications (EuCNC), Jun. 2017.

[31] 5G-PPP White Paper, "5G and the Factories of the Future", [available at] http://clear5g.eu/blog/5g-ppp-white-paper-factories-future, 2015.

[32] Roberto Sabella, ”5G & cloud robotics For industrial IoT”, Ericsson Presentation, Oct. 2015.

[33] Xianjun Jiao, Ingrid Moerman, Wei Liu, and Felipe A. P. de Figueiredo, ”Radio hardware virtualisation for coping with dynamic heterogeneous wireless environments”, EAI International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM), Sept. 2017.

[34] Analog Devices, ”AD-FMCOMMS2/3/4/5-EBZ Zynq and ZED Quick Start Guide”, [available at] https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms2-ebz/quickstart/zynq, Oct. 2016.

[35] Ettus Research, ”List of products”, [available at] https://www.ettus.com/product.

[36] Xilinx, ”Xilinx Zynq-7000 All Programmable SoC”, [available at] https://www.xilinx.com/products/silicondevices/soc/zynq-7000.html

[37] Xilinx, ”Xilinx Zynq-7000 All Programmable SoC ZC706 Evaluation Kit”, [available at] https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html.

[38] Specification, CPRI, “V6.1 Common Public Radio Interface (CPRI); Interface Specification,” NEC Corp., Nortel Networks SA, Siemens Networks GmbH & Co. KG, Ericsson AB, and Huawei Technologies Co Ltd., July, 2014, 129 pages.

[39] T. Wan and P. Ashwood-Smith, ”A Performance Study of CPRI over Ethernet with IEEE 802.1Qbu and 802.1Qbv Enhancements,” 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-6, doi: 10.1109/GLOCOM.2015.7417599.

[40] UHD, ”USRP Hardware Driver”, [available at] https://www.ettus.com/sdr-software/detail/usrp-hardware-driver, retrieved on the 30th of September 2017.

[41] UHD, ”UHD latency test: latency_test.cpp”, [available at] https://github.com/EttusResearch/uhd/blob/maint/host/examples/latency_-test.cpp, retrieved on the 30th of September 2017.

[42] Gomez-Miguelez, Ismael, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano, and Doug J. Leith. ”srsLTE: An open-source platform for LTE evolution and experimentation.” In Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization, pp. 25-32. 2016.

[43] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. 2014. OpenAirInterface: A Flexible Platform for 5G Research. SIGCOMM Comput. Commun. Rev. 44, 5 (October 2014), 33–38. DOI:https://doi.org/10.1145/2677046.2677053

[44] Amarisoft Corp. Website:http://www.amarisoft.com/

[45] National Instruments, ”LabVIEW Communications 802.11 Application Framework”, http://sine.ni.com/nips/cds/view/p/lang/en/nid/213084

[46] WARP Project, ”WARP: Wireless Open Access Research Platform”, http://warpproject.org/trac/

[47] IEEE Standards Association, ”802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks”, Apr. 2016.

[48] Arm Holdings, ”AMBA AXI and ACE Protocol Specification”, developer.arm.com. pp. 121–128. Retrieved 5 July 2019.

[49] Xianjun Jiao, ”Bluetooth Low Energy transmitter and sniffer”, https://github.com/JiaoXianjun/BTLE.

3

A Baseband Wireless Spectrum Hypervisor for Multiplexing Concurrent OFDM Signals

Differently from the previous chapter where we presented a HW-based architecture to virtualize the RF radio HW, here in this chapter, we present a SW-based approach for virtualizing the RF radio hardware.

⋆ ⋆ ⋆

This chapter is based on the homonymous article by

Felipe A. P. de Figueiredo, Ruben Mennes, Irfan Jabandzic, Xianjun Jiao, and Ingrid Moerman.

Published in MDPI Sensors, vol. 20, no. 4, Feb. 2020.

Abstract The next generation of wireless and mobile networks will have to handle a significant increase in traffic load compared to the current ones. This situation calls for novel ways to increase the spectral efficiency. Therefore, in this chapter, we propose a wireless spectrum hypervisor architecture that abstracts a RF front-end into a configurable number of virtual RF front ends. The proposed architecture has the ability to enable flexible spectrum access in existing wireless and mobile

networks, which is a challenging task due to the limited spectrum programmability, i.e., the capability a system has to change the spectral properties of a given signal to fit an arbitrary frequency allocation. The proposed architecture is a non-intrusive and highly optimized wireless hypervisor that multiplexes the signals of several different and concurrent multi-carrier-based radio access technologies with numerologies that are multiple integers of one another, which are also referred in our work as radio access technologies with correlated numerology. For example, the proposed architecture can multiplex the signals of several Wi-Fi access points, several LTE base stations, several WiMAX base stations, etc. As it is able to multiplex the signals of radio access technologies with correlated numerology, it can, for instance, multiplex the signals of LTE, 5G New-Radio (5G-NR) and NB-IoT base stations. It abstracts a RF front-end into a configurable number of virtual RF front ends, making it possible for such different technologies to share the same RF front-end and consequently reduce the costs and increasing the spectral efficiency by employing densification, once several networks share the same infrastructure or by dynamically accessing free chunks of spectrum. Therefore, the main goal of the proposed approach is to improve spectral efficiency by efficiently using vacant gaps in congested spectrum bandwidths or adopting network densification through infrastructure sharing. We demonstrate mathematically how our proposed approach works and present several simulation results proving its functionality and efficiency. Additionally, we designed and implemented an open-source and free proof of concept prototype of the proposed architecture, which can be used by researchers and developers to run experiments or extend the concept to other applications. We present several experimental results used to validate the proposed prototype. We demonstrate that the prototype can easily handle up to 12 concurrent physical layers.

3.1 Introduction

According to [1], the average mobile connection speed in 2016 was 6.8 Mbps and it is forecast to grow at a Compound Annual Growth Rate (CAGR) of 24.4%, reaching nearly 20.4 Mbps by 2021. By 2020 it is forecast that there will be 11.6 billion mobile-connected devices, including M2M modules [1]. Reports like [2] state that future mobile networks will need to support 1 million connections per square kilometer and up to a total of 100 billion connections in total.

The research and development of future mobile networks (i.e., next generation mobile networks) has been derived by several novel breakthroughs in a myriad of areas. The recent advancements of technologies such as licensed assisted access, carrier aggregation, massive multiple-input-multiple-output (mMIMO), co-operative spectrum sensing, beamforming techniques, compressive sensing, machine learning, etc., have provided exciting and encouraging approaches to address the forecast increase in number of connections and their respective speeds [3–7]. These demanding requirements pose daunting challenges to the design of both base stations and user equipment. In particular, at the user equipment side, these new requirements bring a series of daring implementation issues due to limited design budget and hardware resources imposed to the implementation of those devices [6, 8].

To support the expected growth in capacity, the forecast for the future mobile networks, the area throughput, i.e., bit/s/km^2, needs to be increased. Some ways to improve area throughput in cellular networks include densification of existing cellular networks using extra-small cells, provision of peer-to-peer (P2P) communications, multi-tier heterogeneous networks, full-duplex communication, mMIMO, millimeter-wave technologies, cognitive radios, beam division multiple access (BDMA), C-RAN and wireless networks virtualization (WNV) [3–6].

One of the key features to achieve the forecast high data rates is the dense deployment of RRHs. A RRH is basically an RF front-end spatially separated from the baseband processing hardware and encompasses the BS's RF circuitry plus analog-to-digital/digital-to-analog converters and up/down converters. It is generally connected to the baseband processing through a high bandwidth link (e.g., fiber). It is important to point out that the RRH terminology is mainly adopted in 3GPP standards. The dense deployment of RRHs can be achieved, with a relatively lower cost when compared to the deployment of several physical RRHs, by multiplexing several different signals at a single RRH and transmitting them over several virtual RRHs. Carrier aggregation is one of the usage examples, where a BS increases its capacity by allocating more spectrum bandwidth. Infrastructure sharing is another usage example, where densification is achieved by sharing already deployed pieces of equipment, such as the RRH, among several wireless or mobile networks [8, 9].

Infrastructure sharing has the advantage of reducing capital expenditure. It is well known that the deployment of cellular or wireless networks is expensive, and raising the capital for that effort is quite difficult as the operators always want to make the most out of the already deployed infrastructure. Therefore, in order to obtain a better return on the costs related to installation and maintenance of mobile or wireless infrastructure, it would be less expensive to share the already (or newly) deployed infrastructure with other operators than to build overlapping and concurrent infrastructure [10].

This way, RF front-end virtualization becomes very useful for flexible spectrum access, once it allows several PHYs layers to concurrently share a single physical wide-band RF front-end. Some direct consequences of RF front-end virtualization are the reduction in the required physical space for deployment (only a single wide-band RRH is required), energy consumption and price (reduction in redundancy) for the infrastructure provider.

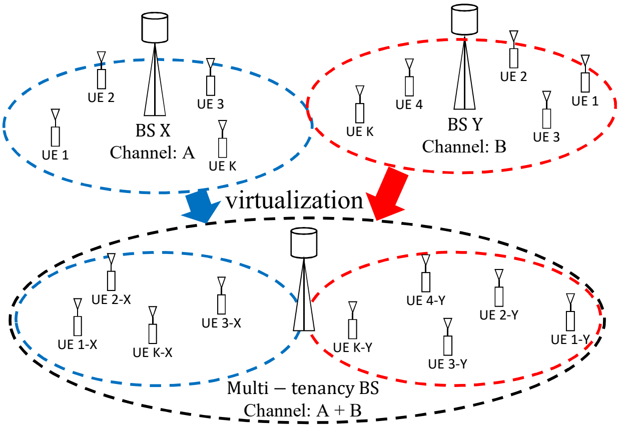


Figure 3.1: One exemplary use case for RF front-end virtualization: A single BS that emulates multiple virtual BSs. Here, subscribers from different networks associate with their corresponding virtual BS by using the same underlying infrastructure.

Figure 3.1 shows an example of a use case where RF front-end virtualization is used in a multi-tenancy application. In this example, two BSs, which are deployed at the same location, use different channels and have totally independent infrastructure (i.e., equipment), are replaced by a single multi-tenancy BS that emulates, through RF front-end virtualization, two BSs where subscribers from the two different networks associate with their corresponding virtual BS.

Another way of achieving such demanding rates is through better use of the available and already underused spectrum bands. In general, due to the presence of primary and secondary users in shared licensed bands and of competing (i.e., opportunistic) users in unlicensed bands, the available spectrum for users in a cognitive radio environment is fragmented and its use is intermittent, i.e., the available spectrum is split into non-contiguous chunks and is not used all the time. This intermittent and fragmented spectrum availability calls for a flexible and agile transmission scheme of the desired signals [11]. The concurrent transmission of several narrow OFDM-based signals allows for the selective use of the available chunks of spectrum, which enhances the spectrum use, consequently improving the area throughput.

Therefore, in this work, we propose a non-intrusive and highly optimized wireless hypervisor architecture for SDRs that ensures coexistence, isolation, and programmability for multi-carrier-based systems (i.e., OFDM) with numerologies that are multiple integers of one another, i.e., systems that have subcarrier spacings that are integer multiples of the smallest adopted frequency spacing and CP lengths that are integer multiples of the maximum possible number of multiplexed signals. The proposed architecture is meant for multiplexing concurrent signals of the same RAT or correlated RATs (e.g., LTE/LTE, LTE/NB-IoT, LTE/5G-NR, 5G-NR/5G-NR, NB-IOT/NB-IoT, Wi-Fi/Wi-Fi, ZigBee/ZigBee, etc.). We focus on supporting OFDM-based systems once such waveform is still one of the most used and important ones being used even in 5G standards like the 5G-NR ones [12, 13]. Therefore, we understand that OFDM will still be widely adopted and employed for a long time.

The proposed hypervisor is a generic architecture that can be used to serve not only Wi-Fi, LTE and NB-IoT-based sensors, i.e., Machine-type communication (MTC) devices, but also serve other kinds of Human-type communication (HTC) devices. The main idea behind the hypervisor is the ability of having several Wi-Fi routers, LTE/NB-IoT or any other OFDM-based RAT base stations sharing the same RF front-end.

We show that with the proposed architecture, spectrum programmability (i.e., the ability to program/change frequency, bandwidth and gain settings) can be decoupled from PHY layer processing and delegated to a virtualization layer, (i.e., the wireless hypervisor) which is added between a set of vPHYs and the hardware/-physical RF front-end. The proposed solution can be applied to both approaches mentioned above of achieving higher area throughput.

We demonstrate that spectrum allocation can be decoupled from the PHY (or baseband processing) layer. In this way, the proposed hypervisor architecture supports flexible spectrum bandwidth allocation by creating a new layer located between several vPHYs and the physical RF front-end. The hypervisor dynamically maps the modulated signals of several vPHYs into configurable chunks of spectrum, before sending the resulting multiplexed signal of the vPHYs to the RF front-end. The hypervisor layer, which can be seen as a spectrum mapping/allocation layer, abstracts the underlying spectrum bandwidth dynamics and provides the vPHYs with a contiguous or non-contiguous (depending on the application of the proposed architecture) set of frequency subcarriers (i.e., virtual spectrum bandwidth), where the desired spectrum bandwidth can be predefined (by operators for instance) or requested in an online basis by the vPHY itself.

Additionally, we present an open-source and free proof of concept prototype that was developed to assess the performance of proposed spectrum hypervisor architecture in real-world experiments. The source code is available at [14] and can be used by researchers and developers to easily run experiments or extend the concept to other applications.

The remainder of the work is organized as follows. In Section 3.2, we present some aspects of hardware virtualization. In Section 3.3, we describe, mathematically, how we multiplex several OFDM signals. Next, in Section 3.4, we list and discuss some important use cases where the proposed architecture can be employed. In Section 3.5 we present and compare some related pieces of work on virtualization. Then, in Section 3.6, we describe the proof of concept prototype developed to assess the feasibility of the proposed architecture. In Section 3.7, some simulation results are presented showing the performance of the proposed architecture. Next, in Section 3.8, we present several experimental results obtained with the prototype. Finally, in Section 3.9, we conclude our work with some conclusions and future work.

3.2 Hardware Virtualization

Hardware (in our case, the RF front-end) virtualization is achieved by means of hypervisors [15, 16]. A hypervisor is a hardware virtualization technique that abstracts, i.e., isolates, multiple concurrent software radio protocol stacks, also known as SDRs, of RATs with correlated numerology from the underlying radio hardware i.e., RF front-end. It allows multiple radio stacks to run on top of a single piece of radio hardware at the same time, i.e., concurrently, such that each protocol stack appears to have its own RF front-end (virtual RF front-end) that can be operated independently. One advantage is that wireless hypervisors help to maximize the effective use of the deployed infrastructure (i.e., servers, RF front ends, etc.).

The hypervisor is the mechanism allowing the seamless sharing of a particular resource by meeting three key requirements: abstraction, programmability, and isolation. The proposed solution addresses each of these requirements as shown in Figure 3.2 and briefly described next.

- **Abstraction**: this feature hides the underlying hardware characteristics and establishes simplified interfaces for accessing and sharing the hardware resources. This feature allows the clients of the hypervisor architecture to use it with no change to their upper layer radio stack, i.e., from the MAC layer upwards. As shown in Figure 3.2, the proposed architecture provides concurrent access to multiple virtual RF front ends, which are exposed to clients through several vPHYs. The proposed architecture can be supported by inexpensive RF front ends (e.g., commodity SDR equipment) or expensive wide-band RRHs. The wireless hypervisor ensures that multiple vPHYs can concurrently coexist on top of the same physical RF front-end. The vPHYs are the point of access of the radio stacks to the virtual RF front ends. Along with the hypervisor, the vPHYs provide a computationally efficient way of multiplexing several baseband signals into a wide-band signal, which consequently splits the spectrum bandwidth provided by the physical front-end. The proposed architecture supports the operation of multiple concurrent vPHYs, implementing totally different air-interfaces (as long as they are OFDM-based signals and have subcarrier spacings that are integer multiples of the smallest adopted spacing) with diverse processing constraints,

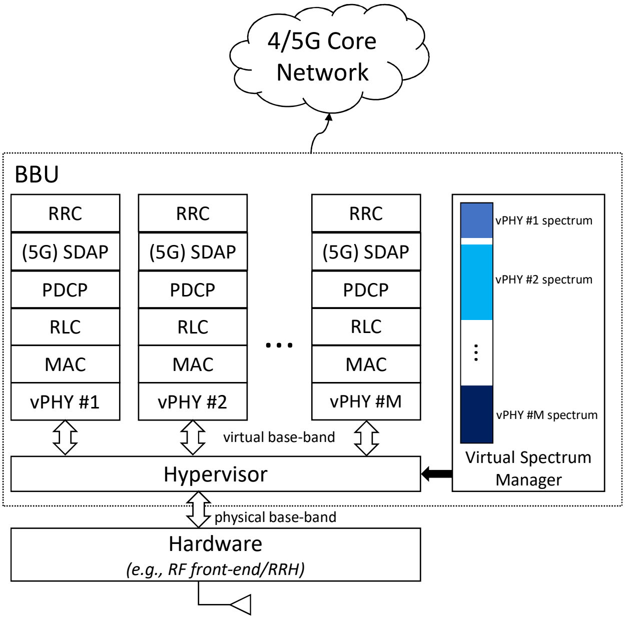


Figure 3.2: High-level view of a hardware hypervisor in 4G or 5G-like networks.

channel bandwidths, medium access schemes, etc.

- **Programmability**: the proposed wireless hypervisor has to provide the multiple radio stacks running on top of it the same level of programmability (or configurability) provided by the single physical RF front-end. This feature is addressed by allowing the multiple radio stacks to configure different center frequencies, bandwidths and transmission/reception gains for individual virtual RF front ends, i.e., the virtual RF front ends must provide the same set of functionalities provided by the physical RF front-end.

- **Isolation**: the proposed wireless hypervisor must make sure that any configuration or wrong configuration does not affect or cause interference to any other collocated radio stack. Isolation is the fundamental requirement that guarantees fault tolerance, security, and privacy to the multiple radio stacks running (i.e., coexisting) on top of the same physical RF front-end [16]. This feature is enforced across multiple clients by providing them predefined access-points and bandwidths that do not overlap.

Figure 3.2 shows one out of several possible multi-tenancy uses for the proposed architecture, where it is employed in 4G or 5G-like mobile networks. Our proposed approach is generic enough to be applied to both 4G or 5G standards or even to a mix of the two standards, since they both use OFDM-based PHYs.

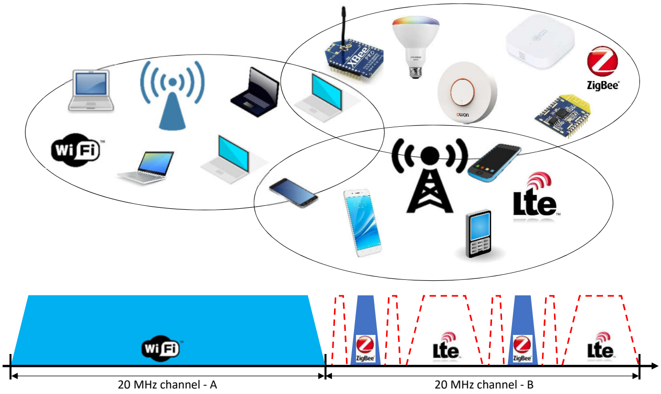


Figure 3.3: Flexible and dynamic spectrum access example. Communications spectrum bandwidths can be flexibly and dynamically changed according to the availability of spectrum as well as application requirements.

However, it could also be, for example, employed by Internet Service Providers (ISP) aiming at sharing their already deployed Wi-Fi access-points (or hot-spots) at locations such as airports, cafes and common shopping areas.

Figure 3.3 depicts yet another use case example regarding dynamic and flexible spectrum access. In the current wireless communications networks, channels have fixed central frequencies and bandwidths. Such static channel allocation causes spectrum fragmentation, lowering the spectrum use efficiency. Figure 3.3 shows a common coexistence scenario in industrial, scientific and medical (ISM) bands. In these bands it is common to have the coexistence of several different technologies, such as Wi-Fi, IEEE 802.15.4 (e.g., ZigBee), LTE-U, etc. [17]. In the figure, two IEEE 802.15.4 networks occupy two channels that overlap with a 20 MHz-wide Wi-Fi channel. As can be seen, the two IEEE 802.15.4 networks may render the channel unusable for Wi-Fi communications due to the narrow-band interference caused by the two IEEE 802.15.4 networks. However, the remaining fragmented spectrum in that channel could be used by a flexible and dynamic system, as depicted by the red-dashed lines in Figure 3.3. As it is shown, the spectrum gaps created by the IEEE 802.15.4 transmissions could be used to establish LTE-like networks (e.g., LTE-U, 5G-NR, NB-IoT, etc.) or any other kind of OFDM-based network. This flexible and dynamic spectrum usage completely removes the concept of predefined and static channel allocations. Based on this approach, frequencies and bandwidths can be dynamically allocated based on the availability of spectrum and the requirements of the applications.

3.3 Multiplexing Concurrent OFDM Signals

In this section, we present a mathematical analysis of the working of the hypervisor proposed in this work and depicted in Figure 3.4. It is important to point out that there is no novelty in the following analysis. However, although being trivial, we haven't found any other piece of work with a similar derivation we could reference. Moreover, we wanted to show how simple and beautiful are the mathematics involved and have a further analysis from the related works on virtualization since none of them have a mathematical analysis like the one we present.

For this analysis, we consider that complex modulated samples, s_i, carrying information are grouped into several narrow sub-bands, $B_k(i)$, which are composed of N_{vPHY} subcarriers. The complex samples, s_i, are symbols drawn from a digital modulation scheme such as BPSK, QPSK, 16QAM, etc. A maximum number of $M = N/N_{\text{vPHY}}$ narrow sub-bands, containing N_{vPHY} subcarriers can be used, where N is the length of the Inverse Fast Fourier Transform (IFFT) used in the hypervisor and c is known as the floor operator and gives the largest integer value that is less than or equal to c. Out of the N_{vPHY} subcarriers, a narrow sub-band might have, N_{vPHY}^{U} carrying useful information, i.e., the modulation symbols s_i, one Direct Current (DC) subcarrier, N_{vPHY}^{DC}, which is set to zero, and N_{vPHY}^{Null} subcarriers that are also set to zero. The nulled subcarriers can be used as frequency guard bands at both edges of the narrow sub-band, allowing for the straightforward realization of anti-aliasing filters. On the other hand, the DC subcarrier allows the use of the simpler and cheaper direct-conversion, also known as zero intermediate-frequency (IF), RF front ends. The narrow sub-band signal can be defined as

$$B_k(i) = \begin{cases} 0, & i = 0, \\ s_i, & 1 \leq i \leq \frac{N_{\text{vPHY}}^{U}}{2}, \\ 0, & (\frac{N_{\text{vPHY}}^{U}}{2}) + 1 \leq i \leq (\frac{N_{\text{vPHY}}^{U}}{2}) + N_{\text{vPHY}}^{Null}, \\ s_i, & (\frac{N_{\text{vPHY}}^{U}}{2}) + N_{\text{vPHY}}^{Null} + 1 \leq i \leq N_{\text{vPHY}} - 1. \end{cases} \tag{3.1}$$

For the sake of clarity, we consider that N is split into M sub-bands with N_{vPHY} subcarriers and that the subcarrier mapping block in the hypervisor maps complex samples into contiguous sub-bands. Please note that the mapping of the M sub-bands, $B_k(i)$, into the IFFT subcarriers/bins corresponds to a frequency translation (mixing) of the sub-band signal, which is centered around 0 [Hz], to a frequency offset defined by the bin number times the subcarrier spacing, $\Delta_f = R_{\text{hyper}}/N$, where R_{hyper} is the sampling rate used by the hypervisor. The hypervisor's sampling rate, R_{hyper}, defines the amount of physical spectrum bandwidth that it can virtualize.

The set, $S(i)$, containing the concatenation of all the M narrow sub-bands, $B_k(i)$, and of length N is transformed into time domain using an IFFT of size N

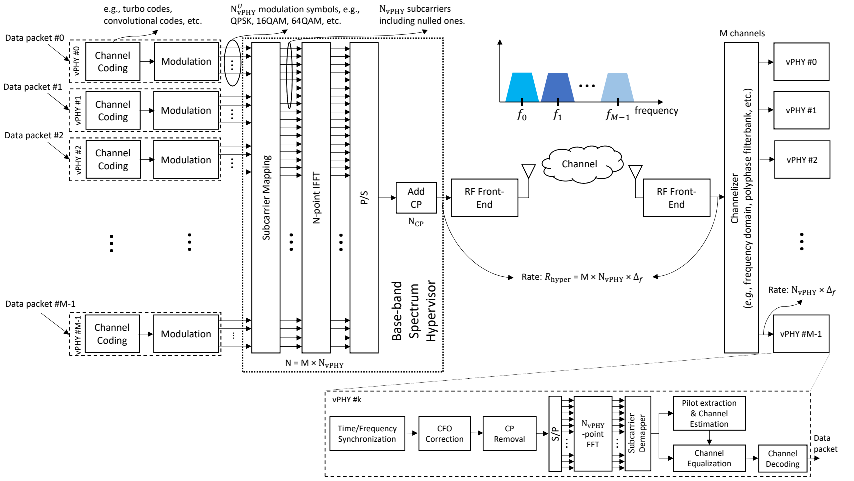


Figure 3.4: Architecture of the proposed baseband wireless hypervisor.

as

$$
\begin{aligned}
s(n) &= \frac{1}{N}\sum_{i=0}^{N-1} S(i)e^{-j2\pi in/N} \\
&= \frac{1}{N}\sum_{k=0}^{M-1}\sum_{i'=0}^{N_{\text{vPHY}}-1} B_k(i')e^{-j2\pi in/N},
\end{aligned}
\tag{3.2}
$$

where $i = i' + kN_{\text{vPHY}}$. Then, based on (3.2), the transmitted wide-band signal, $y_{\text{wBB}}(n)$, including a gain factor and the appending of a cyclic prefix can be represented by

$$
\begin{aligned}
y_{\text{wBB}}(n) = \frac{1}{N}\sum_{k=0}^{M-1}\sum_{i'=0}^{N_{\text{vPHY}}-1} \rho_k B_k(i')e^{-j2\pi(i'+kN_{\text{vPHY}})n/N}, \\
-N_{CP} \leq n \leq N-1,
\end{aligned}
\tag{3.3}
$$

where ρ_k is the frequency amplification factor (or frequency gain), varying from 0 to 1, applied individually to each sub-band and N_{CP} is the CP length, which has to be an integer multiple of M. As can be noticed, $y_{\text{wBB}}(n)$ is a wide-band signal containing all the M multiplexed vPHY signals. As the final step in the transmission chain, the wide-band signal $y_{\text{wBB}}(n)$ is sent to the RF front-end, which will translate the signal into the desired pass-band frequency.

At the receiver side, the wide pass-band signal is translated into a wide baseband signal by the receiver RF front-end. Here we do not consider any channel influence, i.e., the transmitted signal does not suffer from any impairment, and consequently, it is perfectly received at the receiver side. Next, after the frequency translation, the received wide baseband signal $y^r_{\text{wBB}}(n) = y_{\text{wBB}}(n)$, which contains the M multiplexed sub-band (narrow-band) signals, is digitally translated/split into M narrow baseband signals (each one centered around 0 [Hz]), bandwidth restricted by low-pass digital FIR filters, and finally, submitted to a sample rate reduction. These steps are depicted in Figure 3.5, which shows the architecture of a conventional channelizer as described in [18]. As shown in the figure, the output of the k-th channel, which is denoted by $y^{r_k}_{\text{wBB}}$, prior to the down-sampling, is a convolution as defined by

$$\begin{aligned} y^{r_k}_{\text{wBB}}(n) &= \left[y^r_{\text{wBB}}(n)e^{j\theta_k n}\right] h(n) \\ &= \sum_{l=0}^{N_{\text{FIR}}-1} y^r_{\text{wBB}}(n-l)e^{j\theta_k(n-l)}h(l), \\ 0 \leq n &< N + N_{\text{FIR}} - 1 \end{aligned} \tag{3.4}$$

where N_{FIR} is the length of the FIR filter, $\theta_k = \frac{2\pi\delta_k n}{N}$ is the angle corresponding to the digital frequency offset, and δ_k is the center frequency of the k-th narrow sub-band, B_k, in number of subcarrier spacings, Δ_f. After substituting (3.3) into (3.4) we have

$$\begin{aligned} y^{r_k}_{\text{wBB}}(n) = \sum_{l=0}^{N_{\text{FIR}}-1} \frac{1}{N} \sum_{k=0}^{M-1} \sum_{i'=0}^{N_{\text{vPHY}}-1} \rho_k B_k(i') e^{\frac{-j2\pi\Theta_{i',k}(n-l)}{N}} h(l), \\ -N_{CP} \leq n \leq N-1, \end{aligned} \tag{3.5}$$

where $\Theta_{i',k} = (i' + kN_{\text{vPHY}} - \delta_k) \mod N$. The multiplication of $y^r_{\text{wBB}}(n)$ by $e^{j\theta_k n}$ centers the k-th narrow sub-band, B_k, around 0 [Hz] and the convolution with the low-pass filter makes sure that only the signal components belonging to the k-th narrow sub-band go through it while the other sub-bands are filtered out. This way, (3.5) can be re-written as

$$\begin{aligned} y^{r_k}_{\text{wBB}}(n) = \frac{1}{N} \sum_{i=0}^{N-1} \begin{cases} \rho_k B_k(i) e^{\frac{-j2\pi n i}{N}}, & 0 \leq i \leq \frac{N_{\text{vPHY}}}{2}, \\ \rho_k B_k(i) e^{\frac{-j2\pi n i}{N}}, & N - \frac{N_{\text{vPHY}}}{2} + 1 \leq i \leq N-1, \\ 0, & \text{otherwise}, \end{cases} \\ -N_{CP} \leq n \leq N-1. \end{aligned} \tag{3.6}$$

As can be seen by inspecting (3.6), out of N subcarriers/bins, only the lower and upper $\frac{N_{\text{vPHY}}}{2}$ bins contain useful values, i.e., values different from zero. We consider in (3.6) that the filters have a perfect window-shaped frequency response, which is not true in practice but can be approximated by well-designed high order low-pass filters [19].

Next, the wide baseband representation of the k-th narrow sub-band signal, $y^{r_k}_{\text{wBB}}(n)$, is subject to a down-sampling operation, as depicted in Figure 3.5, in

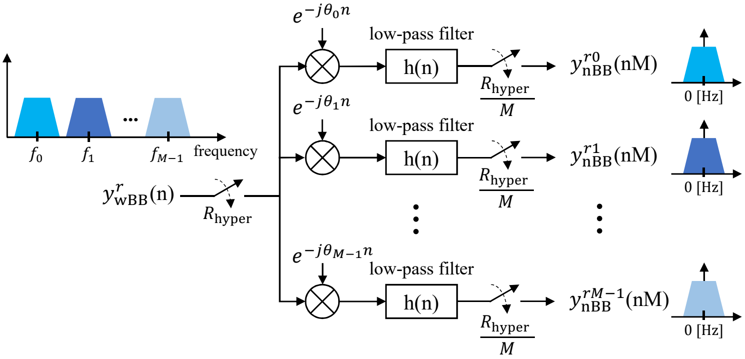


Figure 3.5: Architecture of a conventional channelizer: frequency offset to baseband, low-pass filters, and down-samplers.

order to have a narrow baseband representation of the sub-band signal, which can then be fed into a PHY for data decoding. After down-sampling, the signal expressed by (3.6), can be written as

$$y^{r_k}_{\text{wBB}}(nM) = \frac{1}{N}\sum_{i=0}^{N-1} F(i)e^{\frac{-j2\pi ni}{N/M}}, \quad -N_{CP} \leq nM \leq N-1. \tag{3.7}$$

where the function $F(i)$ is defined as

$$F(i) = \begin{cases} \rho_k B_k(i), & 0 \leq i \leq \frac{N_{\text{vPHY}}}{2}, \\ \rho_k B_k(i), & N - \frac{N_{\text{vPHY}}}{2} + 1 \leq i \leq N-1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

Next, after the down-sampling operation the complex periodic signal $e^{\frac{-j2\pi ni}{N}}$ has its periodicity reduced from N to N/M samples, $e^{\frac{-j2\pi ni}{N/M}}$, and therefore, we can rewrite (3.7) as

$$\begin{aligned} y^{r_k}_{\text{wBB}}(nM) &= \frac{1}{N}\left[\sum_{i=0}^{N_{\text{vPHY}}-1} F(i)e^{\frac{-j2\pi ni}{N/M}}\right. \\ &\left. + \sum_{i=N_{\text{vPHY}}}^{2N_{\text{vPHY}}-1} F(i)e^{\frac{-j2\pi ni}{N/M}} + \cdots + \sum_{i=(M-1)N_{\text{vPHY}}}^{MN_{\text{vPHY}}-1} F(i)e^{\frac{-j2\pi ni}{N/M}}\right] \\ &= \frac{1}{N}\sum_{i=0}^{N_{\text{vPHY}}-1}\left[\sum_{m=0}^{M-1} F(i+mN_{\text{vPHY}})\right] e^{\frac{-j2\pi ni}{N/M}}, \\ &\qquad -N_{CP} \leq nM \leq N-1. \end{aligned} \tag{3.9}$$

By recalling the definition of $F(i)$ given by (3.8), we realize that only half (i.e., $\frac{N_{\text{vPHY}}}{2}$) of the first and last terms of the summation in (3.9) are different from 0, and therefore, it can be re-written as

$$\begin{aligned} y_{\text{wBB}}^{r_k}(nM) &= \frac{1}{N}\sum_{i=0}^{\frac{N_{\text{vPHY}}}{2}} F(i)e^{\frac{-j2\pi ni}{N/M}} \\ +\frac{1}{N}\sum_{i=\frac{N_{\text{vPHY}}}{2}+1}^{N_{\text{vPHY}}-1} & F\left(i+(M-1)N_{\text{vPHY}}\right)e^{\frac{-j2\pi ni}{N/M}}, \\ & -N_{CP} \leq nM \leq N-1. \end{aligned} \tag{3.10}$$

Finally, by using (3.8) and remembering the fact that $N/M = N_{\text{vPHY}}$, we show that the time-domain representation of the k-th narrow sub-band, $B_k(i)$, is recovered by the k-th branch of the channelizer (see Figure 3.5)

$$\begin{aligned} y_{\text{nBB}}^{r_k}(n) &= \frac{1}{N}\sum_{i=0}^{N_{\text{vPHY}}-1} \rho_k B_k(i)e^{\frac{-j2\pi ni}{N_{\text{vPHY}}}}, \\ & -N_{CP}^{\text{vPHY}} \leq n \leq N_{\text{vPHY}}-1, \end{aligned} \tag{3.11}$$

where $N_{CP}^{\text{vPHY}} = N_{CP}/M$ is the cyclic prefix length of the vPHYs. In 3.4, N_{CP}^{vPHY} is the cyclic prefix at the receiver side, just after the channelizer, synchronization and CFO estimation/correction. It is the cyclic prefix for each one of the vPHYs.

It is important to mention that this analysis is the same for any other narrow sub-band centered at any of the N subcarriers of the hypervisor's IFFT module. Additionally, it is also important to notice that each one of the narrow sub-bands, which are multiplexed by the hypervisor, can have different widths. The only requirement is that the width of individual sub-bands is a divisible multiple of N. In this case, if a device is receiving more than one of the multiplexed narrow sub-band signals, then, it has to employ a non-uniform channelizer [20], as polyphase filter-bank channelizers only extract equally spaced spectrum chunks.

3.3.1 Discussion

In this section, we discuss the proposed architecture, its functionalities, features, possible uses and limitation.

As mathematically shown earlier, the proposed wireless hypervisor provides a virtual and discretized (in steps of $\Delta_f = R_{\text{hyper}}/N$) baseband spectrum abstraction layer to the several vPHY layers sitting on top of it. Its key function is the multiplexing, in the frequency domain, of several narrow baseband signals. It receives concurrent sets of modulated signals (e.g., BPSK, QPSK, M-QAM, etc.) from several vPHYs and maps these sets into continuous or non-contiguous subcarriers (i.e., spectrum) that will then be converted into time-domain by the N-point IFFT module and have a proper N_{CP} samples long CP added to it. The digital signal processing carried out by the spectrum hypervisor transforms the sets of modulated vPHY signals into a wide baseband waveform signal that is appropriate for transmission, while keeping the concurrent sets of vPHY signals unchanged and isolated from each other.

The hypervisor supports data flows from multiple concurrent vPHYs and provides each one of them a virtual RF front-end, which can have the following settings configured independently: frequency-domain gain (ρ_k), frequency location (given by the mapping of the sub-band into the hypervisor's IFFT) and bandwidth of the vPHY (N_{vPHY}). Regarding the gain settings for each vPHY, as there is only one physical RF amplifier, the gain of the amplifier is set to a reasonable level (a level that avoids saturating the signal) and the independent vPHYs can set what we called frequency gain (ρ_k), which is a gain applied to the frequency-domain signal and corresponds to a percentage (i.e., 0 up to 100 %) of the RF amplifier's current gain.

Access to the wireless hypervisor is exposed through the vPHYs. The vPHYs work the same way as regular OFDM-based PHY layers (e.g., Wi-Fi, WiMAX, LTE, 5G-NR, NB-IoT, etc.) being the only exception the removal of the OFDM modulation part. In the proposed architecture (see Figure 3.4), OFDM modulation is now executed by the wireless hypervisor, which efficiently multiplexes the baseband signal of several vPHYs into a single wide baseband signal. By removing and transferring the OFDM modulation of all PHYs to the hypervisor, we optimize the whole baseband processing performance by avoiding redundant/unnecessary, for example Fast Fourier Transform (FFT) and IFFT, operations that would be required to multiplex several concurrent regular PHY signals into a wide baseband signal [21–23]. The down-side of our approach is that it can only operate with multi-carrier-based (i.e., OFDM) signals.

At the receiver side, different receiving approaches might be followed, depending on how the proposed hypervisor is employed to generate the wide baseband signal. The first approach refers to a communications connection between two devices (i.e., a point-to-point connection). In this approach, as a first step, the received wide baseband signal coming from the RF front-end is split by a channelizer into equally or unequally (i.e., non-uniform) spaced spectrum chunks and then, the down-converted and down-sampled narrow sub-band signals are fed into the narrow-band PHY receivers running on the device. The second approach is similar to the first one, where the only difference is that the signal coming from the RF receiver front-end is not split into narrow sub-bands. Here the wide-band received signal is fed directly into the OFDM demodulator of a corresponding wide-band vPHY. In the third approach, several independent, distributed and narrow-band devices have their RF receiver front ends tuned to the center frequency and transmission bandwidth of each one of the transmitted vPHY signals. The fourth approach represents a mix of the previous three approaches, where there could be wide-band devices tuning to more than one transmitted narrow-band vPHY signal while other narrow-band devices would only be tuning to individual transmitted narrow-band vPHY signals. In all the approaches mentioned above, the signal multiplexing carried out at the transmitter side is transparent to the PHY receivers, meaning that the radio stack at the receivers does not need to be modified.

As a limitation, the proposed hypervisor architecture supports the operation of multiple concurrent RATs as long as they are OFDM-based signals, have subcarrier spacings that are integer multiples of the smallest adopted frequency spacing and CP lengths that are integer multiples of the maximum number of vPHYs, which is given by M, i.e., RATs with numerologies that are multiple integers of one another, which are also referred here as RATs with correlated numerology. This limitation was chosen at design-time so that the proposed architecture could get rid of several digital signal processing tasks (e.g., up/down-conversion, filtering, etc.) necessary by a BS to handle uncorrelated RAT numerologies (e.g., LTE, Wi-Fi, ZigBee, etc.) but that would increase the processing time and complexity of a given solution for multiplexing these RATs. Therefore, we have chosen this limitation for the sake of processing time and complexity optimization. Our approach is novel when compared to other ones (please, refer to Section 3.5) that support the multiplexing of RATs with uncorrelated numerology by performing several additional digital signal processing tasks that affect the processing performance of a BS, especially when the number of multiplexing signals increases. With our proposed approach, we wanted to focus on correlated RATs (i.e., RATs with numerologies that are multiple integers of one another) and with that have an optimized architecture for such RATs. Therefore, based on the correlated numerology limitation, it is not possible to employ the proposed architecture to multiplex the signal of RAT technologies with uncorrelated numerologies such as LTE, Wi-Fi and ZigBee. As mentioned in the use cases given in Section 3.4, the proposed architecture is meant for multiplexing concurrent signals of the same RAT or correlated RATs (e.g., LTE/LTE, LTE/NB-IoT, NB-IOT/NB-IoT, Wi-Fi/Wi-Fi, ZigBee/ZigBee, etc.). Consequently, if the different RATs have correlated numerologies, then, as analytically proved in Section 3.3, these different and concurrent RAT signals can be orthogonally multiplexed successfully.

One could argue why we haven't applied separate up-conversions to each one of the PHYs and then have their signals summed up before transmission. This approach is also possible, but would not be efficient for an SW implementation as the one presented in this work. It would increase processing time and complexity for implementations in SW since it would involve one IFFT, CP addition, and up-conversion per PHY plus the summation of all signals. In HW, it would take more logical elements but would not have a high processing time since we can have processes running in parallel. The proposed architecture eliminates those additional signal processing tasks by mapping the signals of a vPHY directly into the IFFT.

Moreover, keeping the orthogonality between vPHYs could be an issue since we would have different up-converters (i.e., oscillators with possibly different phases).

3.4 Use Cases

In this section, we present and discuss some possible use cases for the proposed architecture.

3.4.1 Dynamic Spectrum Access

The FCC has reported that licensed spectrum bandwidths are greatly underused [24]. One of the examples mentioned in the report is the TV spectrum, which has one of the lowest use rates. That spectrum band is in most cases left totally unused in areas not so populated (e.g., rural areas), and due to the fact that it is referred to as TV whitespace. Based on this under-use of spectrum originally meant only for TV broadcasting, regulators are opening up this previously licensed spectrum for unlicensed use [25]. TV whitespaces for unlicensed use brought about a revolution to cognitive radio, spectrum sensing as well as to dynamic spectrum access [26]. TV whitespaces allow for the opportunistic use of vacant TV channels for data communications [27, 28].

Therefore, based on the opportunistic use of vacant spectrum, one of the use cases of the proposed wireless hypervisor is instantiating it as a NC-OFDM PHY, where one or several vPHYs could, based on spectrum availability/occupancy, have their data mapped into subcarriers that do not interfere/overlap with primary users in a DSA scheme, and thereby enabling efficient use of the spectrum [11]. For this use case, an NC-OFDM PHY node would employ the proposed hypervisor architecture as a wide-band OFDM modulator, where the subcarrier mapping module depicted in Figure 3.4 maps the modulated symbols only into subcarriers that correspond to vacant spectrum, i.e., not being used by the primary or other users. At the receiver side, the signal is demodulated and decoded as a wide-band NC-OFDM signal where some subcarriers had their values set to zero. The PHY receiver for this signal is depicted in the bottom right corner of Figure 3.4, and is nothing but a generic OFDM demodulator/decoder where the subcarrier demapper module needs to know beforehand which subcarriers are active during a given transmission interval [11].

Figure 3.6 depicts a possible snapshot of the spectrum usage in the TV whitespace when the proposed architecture is used as a dynamic packet-based PHY layer with allocated bandwidth being dynamically changed according to vacant spectrum. In this example, the PHY layer establishes concurrent/simultaneous communications with several nodes. As can be seen, the channel's center frequency and allocated transmission bandwidth are changed throughout time, depending on

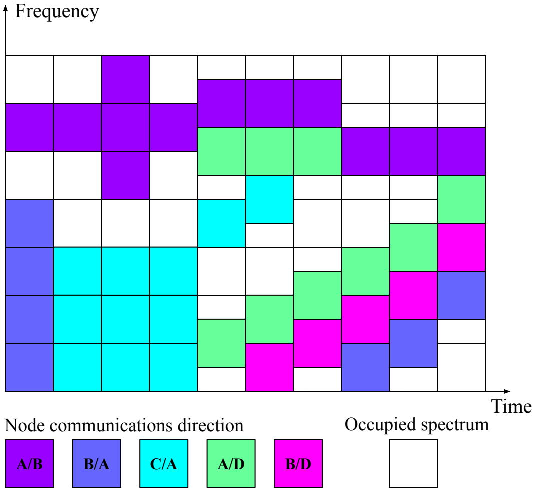


Figure 3.6: Snapshot of the spectrum usage in the TV whitespace.

spectrum availability and/or traffic load.

Another use case for the proposed architecture is in the implementation of a spectrum-sharing scheme between wireless operators and Fixed Satellite Services (FSS) (FSS) in the CBRS band. FSS stations must share spectrum with new entrant wireless operations, while the entrant networks must ensure that the interference that they introduce to the incumbent FSS remains below a specified threshold [29, 30]. The aim with the launch of CBRS in the USA is that wireless systems should dynamically share the spectrum among them [30]. In this use case, the proposed architecture could easily adapt its transmission bandwidth to the available (either continuous or discontinuous) spectrum at any specific time. This use case is similar to the one described by Figure 3.6. Mainly aligned with this use case, DARPA, the Defense Advanced Research Projects Agency from the United States, has started the Spectrum Collaboration Challenge with the aim to encourage research and development of smarter/more intelligent coexistence and collaboration techniques of heterogeneous networks in the same wireless spectrum bands [31, 32]. One of the examples they have been advocating for is the adoption of such spectrum-sharing technologies in the CBRS band [33, 34].

3.4.2 Network Densification

The capacity of an Additive white Gaussian Noise (AWGN) channel is given by the following equation [9]

$$R < C = m\left(\frac{W}{n}\right)\log_2\left(1+\frac{S}{I+N}\right), \tag{3.12}$$

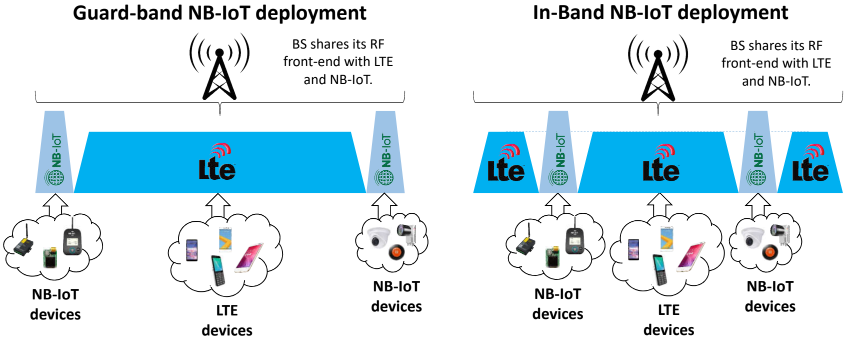

Guard-band NB-IoT deployment
BS shares its RF front-end with LTE and NB-IoT.
NB-IoT
Lte
NB-IoT
NB-IoT devices
LTE devices
NB-IoT devices
In-Band NB-IoT deployment
BS shares its RF front-end with LTE and NB-IoT.
Lte
NB-IoT
Lte
NB-IoT
Lte
NB-IoT devices
LTE devices
NB-IoT devices

Figure 3.7: Application of the proposed hypervisor where a single BS serves both LTE and NB-IoT devices/sensors.

where W is the BS allocated bandwidth, n denotes the BS load factor (i.e., the number of users sharing the given BS), m is the spatial multiplexing factor (i.e., it denotes the number of spatial data streams connecting the BS and devices), S gives the signal power and I and N represent the interference and power noise, respectively, experienced at the receiver side. After analyzing the equation, it is possible to see that the capacity can be increased by decreasing the BS load factor, which can be attained through cell splitting. Cell splitting involves deploying a larger number of BSs and making sure that the user traffic is evenly distributed among all the deployed BSs [9]. A possible consequence of cell splitting/densification is that it might improve the signal power, as the users could now be closer to the new deployed BS and consequently experiencing a reduced path-loss.

The dense deployment of infrastructure is a precondition for cutting the BS load factor n down in (3.12). However, the deployment of additional BSs entails significant costs and detailed site survey/planning [35]. Therefore, another use case, which represents a very important application for the proposed architecture, is instantiating the proposed architecture as an RF front-end multiplexer, where several vPHYs share a single physical RF front-end. This could be employed in multi-tenancy cases [21], where a cellular network infrastructure provider shares their owned infrastructure and/or spectrum with Mobile Virtual Network Operators (MVNOs) or vertical markets such as energy, automotive, city management, food and agriculture, health-care, government, public transportation, manufacturing, etc. In this case, the infrastructure provider shares its already deployed infrastructure, which ranges from BBUs to the RF front ends (also known as RRH). By employing multi-tenancy schemes, operators, MVNOs and verticals can decrease their Capital Expenditure (CAPEX) and OPerational Expenditure (OPEX). Multi-tenancy makes the deployed infrastructure more energy efficient/greener by allowing a reduced number of BS sites, and, therefore, largely reducing the power consumption of air conditioning and other site support pieces of equipment.

The infrastructure provider can, for example, split its owned spectrum band into smaller chunks and lease it to MVNOs or verticals. Another possible example is instead of splitting its own spectrum into smaller chunks, is the possibility of providing other operators access to their own spectrum bands due to the wideband capabilities of the current RRHs, ranging from 10 to 250 MHz of useful instantaneous bandwidth [36–38]. The virtualization of the physical RRHs is an important step in the direction of multi-tenancy networks as being studied by the 3GPP [39].

This possible application/instantiation of the proposed architecture enables the RAN to be made available as a service, also known as RAN as a service (RANaaS), to MVNOs and verticals. vPHYs create new ways for infrastructure providers to monetize their owned spectrum bandwidth and deployed infrastructure. In this way, infrastructure providers can offer vPHYs as a Service (vPHYsaaS) in order to provide isolated and independent virtual networks to MVNOs and/or verticals sitting on the top of a shared physical infrastructure [40]. For this use case, a BBU, providing vPHYs as a service to MVNOs/verticals/operators, would have the proposed hypervisor architecture multiplexing the signal of several vPHY at downlink side while a channelizer would be deployed at uplink side in order to provide each one of the vPHYs with a signal corresponding to its allocated bandwidth, just as depicted in Figure 3.4. This use case is aligned with standardization efforts made by 3GPP that consider a BS serving both LTE and NB-IoT users (mobile devices and sensors as shown in Figure 3.7) [40]. This application of the proposed hypervisor allows a single BS to flexibly serve LTE devices and NB-IoT devices (e.g., sensors), decreasing costs and increasing the spectral efficiency by densification.

3.5 Related Work

In this section, we describe and discuss some related pieces of work on virtualization.

The related works on virtualization can be split into two main categories, time- or frequency-multiplexing. The time-multiplexing approaches achieve virtualization by splitting the access time to a common Wi-Fi access-point PHY layer [41, 42]. This is possible due to the fact that Wi-Fi is a packet-based wireless network, where access points do not transmit data all the time, and therefore, being able to allow virtual Wi-Fi radio stacks to use its idle time. The frequency-multiplexing approaches can be further divided into two sub-categories based on the type of the virtualized resource, which can be the spectrum bandwidth through the virtualization of the RF front-end [21–23, 27, 43] or the time-frequency resource grid through the virtualization of the resource blocks (RB) provided by Orthogonal Frequency Division Multiple Access (OFDMA)-based radio technologies such as LTE and WiMax [44–47].

Out of all the compared related works, only a few are technology agnostic, i.e., can provide virtual radios to any RAT [21–23, 27, 43]. However, the issue with these works is that they exchange optimized performance for being generalized so that they can operate with different RATs. In most of these works [21–23, 43], in order to be agnostic, the hypervisor layer re-implements operations (e.g., FFT, IFFT, Subcarrier Mapper, etc.) that are already performed by the PHY layers of the individual RATs, creating an extra overhead that decreases the performance of the solution.

An important comparison point is the dynamic creation and destruction of virtual radios without interfering with or stopping the hypervisor or other already instantiated virtual radios. Out of all works, only a few do not support such feature [21–23, 27, 44], where the number of instantiated virtual radios and their respective bandwidth allocations must be configured before running the hypervisor.

Another interesting point of discussion is the maximum possible number of instantiated virtual radios concurrently running during the experiments. The compared works were experimentally tested with the number of concurrently running virtual radios ranging from 1 to 4 virtual radios; however, on the other hand, our prototype has been experimentally tested with 12 concurrently running vPHYs.

Finally, it is also important to add that independent gain configuration is an important feature to be exposed to the virtual radios once, for example, devices might be at totally different locations; however, it seems that independent gain configuration is not a major concern to most of the related works, once only a very few mention its support [23, 27, 43].

Differently from the other compared works, the virtualization architecture proposed in this work was designed to be highly optimized for sharing the same underlying physical RF front-end among several concurrent multi-carrier-based virtual radios (i.e., vPHYs). It provides dynamic access to several concurrent and configurable virtual RF front ends (e.g., frequency gain, frequency location, and bandwidth) that are accessed through multiple vPHYs. The vPHYs can be instantiated in real time without interfering with running vPHYs and without the necessity to stop the hypervisor.

The majority of the compared related works do not make their source code available [27, 41, 42, 44–47]; however, we believe that research on this field can only progress if the different implementations are made available for comparison and a better understanding of their functionalities and features. Therefore, we make our proposed architecture prototype available at GitHub [14]. The source code includes some examples to measure the prototype's performance, a Graphical User Interface (GUI) for visualizing the transmitted spectrum and a channel emulator that can be used to run experiments without the necessity of having a dedicated piece of physical RF front-end. It emulates AWGN, Rayleigh and Multi-

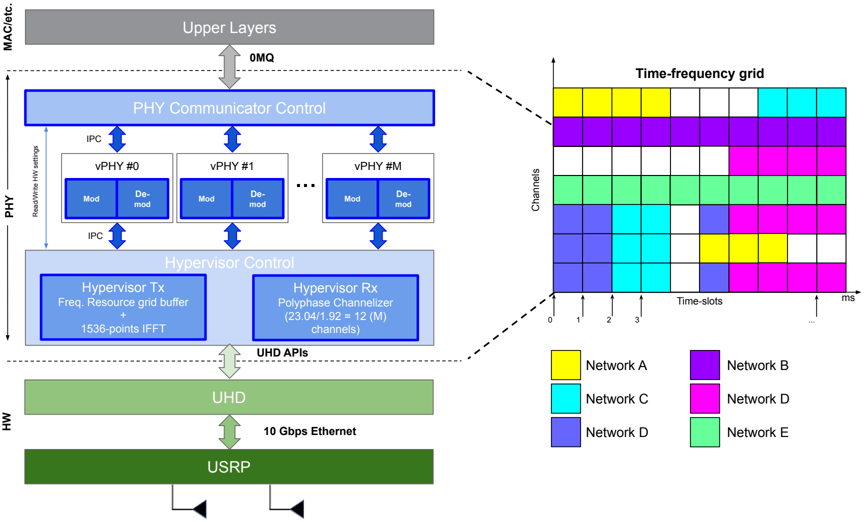


Figure 3.8: High-level architecture of the implemented prototype.

path channels with several different SNR values. The prototype makes extensive use of Single Instruction Multiple Data (SIMD) functions, including the FFT and IFFT implementations [48], making it even more optimized.

Table 3.1 presents a comprehensive comparison of related virtualization and hypervision works.

3.6 Proof of Concept Prototype

In this section, we describe a proof of concept prototype developed to verify the performance of proposed spectrum hypervisor architecture in real-world experiments.

Figure 3.8 depicts the high-level architecture of the implementation of the proposed architecture. As can be seen, the prototype is composed of several modules, namely PHY communicator control, M vPHYs, and the hypervisor control module, which is in turn, composed of the modules hypervisor Tx and Rx.

The prototype is an open-source software-defined PHY layer designed to multiplex and receive (i.e., demultiplex) the signal of several vPHYs [14]. It is implemented based on the srsLTE library [49]. srsLTE is an open-source and free LTE software-based library [49]. The prototype can run on top of several Ettus SDR devices such as the Ettus USRP X family or National Instruments' (NI) RIO SDR devices [50, 51] by using the USRP UHD software API [17, 52]. Therefore, the prototype accesses the SDR device through the UHD driver and its APIs [53].

The communication between the prototype and the upper layers is carried out

Table 3.1: Comparison of related works.

	Related Work									
Comparison Metric	This work	[23]	[43]	[44]	[41]	[45]	[27]	[46]	[47]	[21]	[42]
Multiplexing domain	Frequency	Frequency	Frequency	Frequency	Time	Frequency	Frequency	Frequency	Frequency	Frequency	Time
Virtual resource type	Spectrum	Spectrum	Spectrum	RB	AP in time	RB	Spectrum	RB	RB	Spectrum	AP in time
RAT agnostic	Only MC [10]	Yes	Yes	Only WiMax	Only Wi-Fi	Only LTE	Yes	Only WiMax/LTE	Only LTE	Yes	Only Wi-Fi
Spectrum flexibility [11]	Yes	Yes	Yes	Yes	FC/FB [12]	Yes	Yes	Yes	Yes	Yes	FC/FB [12]
Operation mode	S/PB	S	S	S	PB	S	S	S	S	S	PB
Simulation validated	Yes	No	No	Yes	No	No	No	No	Yes	No	No
Experimentally validated	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Implementation	SW	SW	SW	SW	SW	SW	SW	SW	SW	SW	SW
Create self-interference	No [1]	Yes	Yes	No [1]	No [2]	No	Yes	No	No	Yes	No [2]
Create OOB emissions	Yes	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes
Individual gain control	Yes	Yes	Yes	NA	No	No	Yes	NA	NA	No	No
Open-source prototype	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
Dynamic allocation [3]	Yes	No	Yes	NA	Yes	Yes	No	Yes	Yes	No	Yes
Radio resource isolation [4]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual radio independence [5]	Yes	No	Yes	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Concurrent Tx/Rx at different channels	Yes	Yes	Yes	Yes	No	No [6]	Yes	No [6]	No [6]	Yes	No
Number of VR instances [7]	12	2	2	2 slices [9]	2 [8]	3 slices [9]	3	NA	4 slices [9]	4	2 [8]

Legend: RB: resource-block - MC: multi-carrier - AP: access-point - SW: software - S: streaming - PB: packet-based - NA: information not-available - VR: virtual-radio - FC: fixed channel center frequency - FB: fixed channel bandwidth.

[1] OFDM-based systems do not create self-interference as the subcarriers are mutually orthogonal. [2] Once it time-multiplexes the access-point. [3] Allow on demand destruction and creation of VRs without interrupting the operation of the spectrum hypervisor or other VRs. [4] Allocate non-overlapping sub-bands to different VRs and prevent interference among VRs, e.g., guard bands. [5] Ensure that VRs cannot interfere with the operation and performance of other VRs, even in the case of a malfunctioning or misbehaving VRs. [6] Use the concept of virtual RBs, therefore, concurrent transmissions only happen at different RBs within the same channel. [7] The maximum number of VRs instantiated during the experiments. [8] A single AP PHY that is time-shared between two VRs. [9] A single PHY layer that has its RBs split into slices, creating VRs. [10] Optimized for multi-carrier-based waveforms, e.g., OFDM. [11] Flexibility in setting different channel center frequencies and bandwidths to different concurrent VRs. [12] The approach only works with fixed channel center frequencies and bandwidths as it time-multiplexes a Wi-Fi AP.

Table 3.2: List of messages and their respective parameters.

vPHY Message	Parameter	Type	Unit	Range
Tx control	vPHY ID	uint32	-	0–1
	Tx gain	float	%	0-100%
	Tx vPHY BW	uint8	MHz	1–6 [1]
	Tx channel	uint32	-	≥ 0
	Data struct	uchar[]	-	uchar range
	Modulation Code Scheme (MCS)	uint8	-	0-28
	# of resource blocks	uint8	-	1–100
	User Data length	uint32	-	> 0
	User data	uchar[]	-	uchar range
Rx control	vPHY ID	uint32	-	0–1
	Rx vPHY BW	uint8	MHz	1–6 [1]
	Rx channel	uint32	-	≥ 0
Rx statistics	vPHY ID	uint32	-	0–1
	Channel Quality Indicator (CQI)	uint8	-	0–15
	Received Signal Strength Indication (RSSI)	float	dBW	float range
	Noise	float	dBW	float range
	Decoded MCS	uint8	-	0–28
	cyclic redundancy check (CRC) error counter	uint32	-	≥ 0
	Data length	uint32	-	≥ 0
	Received data	uchar[]	-	uchar range

[1] These numbers correspond to the following LTE bandwidths: 1.4, 3, 5, 10, 15 and 20 MHz respectively.

through a set of three well-defined messages, which are exchanged over a ZeroMQ bus [54], i.e., the prototype and the upper layers are interconnected through a publish-subscribe messaging system known as ZeroMQ [54]. ZeroMQ is a high-performance asynchronous messaging library, designed to be used in distributed or concurrent applications [54]. The set of vPHY messages is designed with Google's Protocol Buffers (protobuf) [55]. Protobuf is used for data serialization and works perfectly with the 0MQ messaging library [54].

The first two vPHY messages, called, Tx and Rx Control, are used to control and configure the transmission and reception of user data, respectively. The parameters in these two control vPHY messages should be configured and sent to the individual vPHYs by the upper layers before the transmission of every new subframe. Each vPHY control message, as the name suggests, controls the operation of only one vPHY. The remaining of the message, called Rx statistics vPHY message, is used to give upper layers feedback on the operation of each individual vPHY.

Tx control messages transport the user data to be transmitted (i.e., TBs) and transmission parameters such as vPHY ID, vPHY Tx BW, MCS, Tx gain, Tx channel, number of resource blocks used by that user, data length, and data. The vPHY ID field is used in all messages to specify to which one of the vPHYs a control message is meant to or received from. **Rx control** messages are used to configure reception parameters such as vPHY ID, vPHY Rx BW, and Rx channel.

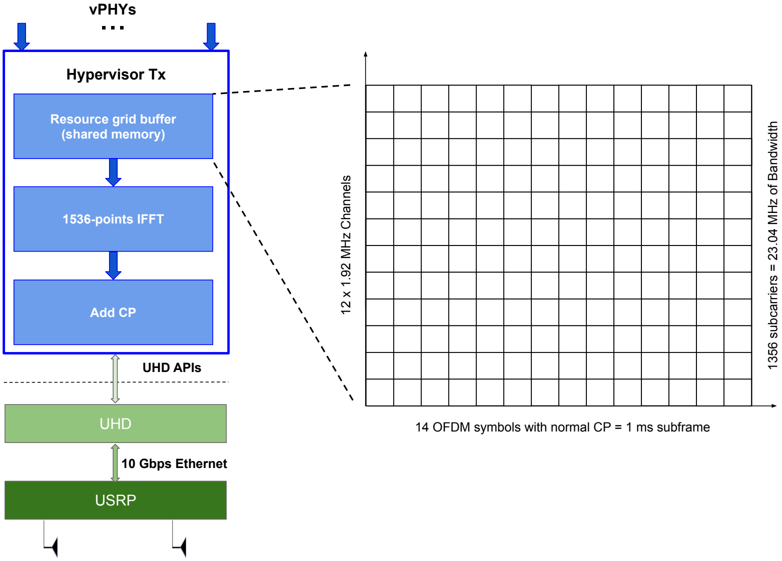

vPHYs
Hypervisor Tx
Resource grid buffer (shared memory)
1536-points IFFT
Add CP
UHD APIs
UHD
10 Gbps Ethernet
USRP
12 x 1.92 MHz Channels
1356 subcarriers = 23.04 MHz of Bandwidth
14 OFDM symbols with normal CP = 1 ms subframe

Figure 3.9: Internal architecture of the Hypervisor Tx module.

Rx statistics messages carry the vPHY ID, received decoded user data, and reception statistics such as RSSI, CQI, decoded MCS, CRC error counter, etc. The vPHY messages and their parameters are summarized in Table 3.2.

Next, we describe each one of the modules composing the prototype.

- **PHY Communicator Control**: this module is responsible for the exchange of messages with several and possibly independent upper layers, e.g., the MAC layers from different users or operators. This module works on a subframe basis, meaning that the connected upper layers always send/receive in one control message to/from the module the content of a subframe as the minimum unit of data exchange. The received control messages, carrying user data, are then relayed to the respective vPHY by using the vPHY ID in the control message. The decoded user data is sent to the respective upper layer also by using the vPHY ID.

- **vPHY**: modulates and demodulates the user data. After modulating the data, each vPHY maps, according to the channel configured in the Tx control message, its N^U_{vPHY} modulated symbols into a memory buffer, called, *resource grid buffer*, which is a 1 ms (i.e., the duration of a subframe) representation of the frequency-domain spectrum band multiplexed by the PHY prototype. Each vPHY only has to map the N^U_{vPHY} data symbols (i.e., useful symbols), while the remaining positions or subcarriers of the buffer have their values already set to zero before every new transmission. In the current implementation of the prototype, $N^U_{\text{vPHY}} = 72$ and $N_{\text{vPHY}} = 128$ subcarriers. The *resource grid buffer* is a discretized, in number of subcarriers or IFFT points, representation of the spectrum for the duration of 1 ms. In this version of the prototype, we used a 1536-point IFFT. The *resource grid buffer* is a memory buffer that is shared by all vPHYs. In the demodulation case, the IQ samples that are output by the Hypervisor Rx module are decoded accordingly by the respective vPHY. In the current implementation of the prototype 12 vPHYs can be instantiated and concurrently transmit/receive their data. This number of vPHYs is obtained by dividing the number of IFFT points, $N = 1536$, by the total number of vPHY subcarriers including the null ones, $N_{\text{vPHY}} = 128$.

- **Hypervisor Tx**: applies a 1536-point IFFT to the *resource grid buffer*, adds CP and transfers the IQ samples to the USRP for transmission over the air. The internal architecture of the Hypervisor Tx module is depicted in Figure 3.9. As showed in the figure, the Hypervisor Tx can be seen as an OFDM modulator where each OFDM symbol is created by reading and processing the consecutive 1536 data symbols stored at the *resource grid buffer*. As showed in Figure 3.9, the *resource grid buffer* stores data of 12 channels $\times$ 14 OFDM symbols, totaling 1 ms of data.

- **Hypervisor Rx**: applies a FIR polyphase filter-bank channelizer to the IQ samples received from the USRP and outputs the M down-converted channels to the vPHYs for data demodulation and decoding. The channels output by the channelizer are centered at 0, $N_{\text{vPHY}} \times \Delta_f$, $2 \times N_{\text{vPHY}} \times \Delta_f$, $3 \times N_{\text{vPHY}} \times \Delta_f, \ldots$, $M - 1 \times N_{\text{vPHY}} \times \Delta_f$, respectively, where $N_{\text{vPHY}} \times \Delta_f$ $= 1.92$ MHz

For improved processing performance, each one of the just described modules runs on an exclusive thread. As the prototype works on a 1 ms basis, it is possible to have a mix of streaming and burst-based transmissions as shown in Figure 3.8.

The only inconvenience found in the proposed architecture is that it requires that all the involved PHY implementations be jointly modified (although large parts of the implementations, suitable for physical RF front ends, can be directly reused). This means that in practice, the proposed approach is only valid when the researchers/developers have access to the source code (either open or proprietary source code) of all involved OFDM-based PHYs, since they need to have their code modified in order to be connected to the hypervisor. However, we would like to point out that presently the open-source community offers several full-blown radio stack projects, being the most important and well-known ones srsLTE [49] and OAI [56] projects. Both of them offer mature, well-organized, well-documented, well-written and easily modifiable open-source code for LTE, NB-IoT and 5G-NR radio stacks, including PHY, MAC, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), Radio Resource Control (RRC), Non-Access Stratum (NAS), S1 Application Protocol (S1AP) and Gateway (GW) layers besides the implementations of EPCs. Additionally, they also have very active mailing lists for researchers/developers to ask questions and report issues with the code.

Due to the way that the cyclic prefix is added at the transmitter side, after the IFFT, all transmissions need to be synchronized. That can be a limiting factor for latency constrained or services that can not wait too long for transmissions due to QoS requirements, for example. In those cases, the processing granularity (i.e., the number of data necessary for a transmission, which is equal to 1 ms today) could be decreased to accommodate those requirements. For technologies that are more tolerant to latency or services with relaxed QoS requirements, the *resource grid buffer* (shared memory) and a mechanism (with semaphores and mutexes) are enough to synchronize transmissions. Therefore, with the implemented prototype, that does not pose a significant limitation since we control and synchronize all vPHYs.

Although using a polyphase filter in our prototype, which is an efficient filter bank implementation, the filter order will impact on both time-processing and decoding performance. If the filter is too short, the reconstructed signal won't be clean, which will affect the BER. If the filter is too long, the processing time will also be too long, which might "clog" the system, i.e., the filtering processing will take more time than the system generates new samples.

3.7 Simulation Results

In this section, we present some simulation results carried out to validate and assess the functionality of the proposed architecture.

For the first validation, we want to check the average Mean Squared Error (MSE) related to the multiplexing of several vPHY modulated signals by the hypervisor. In this simulation, we measure the error between OFDM symbols, y_{PHY}, created with an 128-point IFFT, N_{PHY}, plus 9-sample long CP, N_{PHY}^{CP}, and the OFDM symbols received from the hypervisor after down-conversion and channelization, y_{vPHY}. Each (v)PHY signal modulates 72 consecutive subcarriers, which translates into a useful transmission bandwidth of 1.08 [MHz] when a subcarrier spacing, Δ_f, of 15 [KHz] is used. We average the MSE error over 10^5 iterations, where at each iteration, we have the single PHY and all the vPHYs modulated with the same randomly generated data. For this simulation, we use 12 vPHYs, i.e., $M = 12$, where each vPHY has its signals mapped into 128 consecutive subcarriers, N$_{\text{vPHY}}$, totaling 1536 subcarriers, which is the number of points used by the IFFT block in the hypervisor, N. At the receiver side, we use a polyphase FFT analysis filter-bank [18] to split the wide-band signal into multiple uniformly

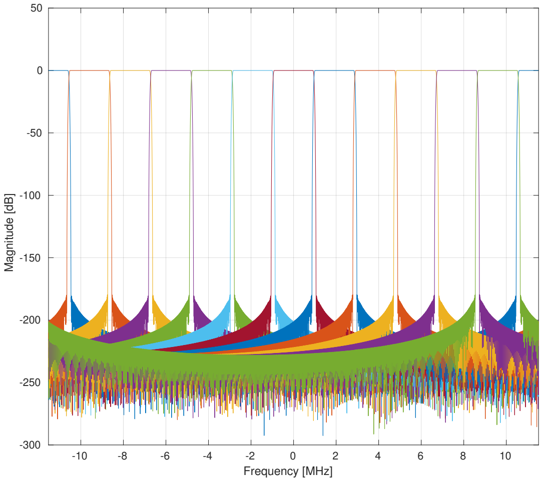


Figure 3.10: Frequency response of a polyphase filter-bank with stop-band attenuation of 180 *[dB] and* 512 *filter coefficients per sub-band.*

spaced narrow sub-bands. It has a 180 [dB] stop-band attenuation and 512 filter coefficients per sub-band.

The frequency response of the polyphase filter-bank used in all the simulations presented in this section is depicted in Figure 3.10. As can be seen in the figure, the physical spectrum band, R_{hyper}, which is provided by the RF front-end, is split into M equal-bandwidth equally spaced sub-bands of 1.92 [MHz] (i.e., $\frac{N \times \Delta_f}{M}$). As can be seen, in the figure, due to the frequency response of the polyphase filter-bank one channel is split over the most negative and positive frequencies. In practical implementations, as in our prototype's case, the received multiplexed signal is shifted so that the each sub-band is exactly centered at the channelizer's expected center frequencies. It is important to mention that the stop-band attenuation and filter order parameters play an important role in the fidelity of the multiplexed signals to the single PHY one [18].

The MSE for the k-th vPHY is calculated as defined by (3.13) below.

$$\text{MSE}_k = \mathbb{E}\left[\frac{1}{(N_{\text{PHY}} + N_{\text{PHY}}^{CP})} \times \sum_{n=0}^{(N_{\text{PHY}}+N_{\text{PHY}}^{CP})-1} |y_{\text{PHY}}(n) - y_{\text{vPHY}}(n)|^2\right]. \tag{3.13}$$

Additionally, we have also calculated the average Modulation Error Ratio (MER) to compare the error introduced by the multiplexing to the vPHY modulated signal. The MER compares the error between the modulated data symbols (i.e., the

Table 3.3: MSE and MER for several different modulation schemes.

	Modulation Order					
	BPSK	**QPSK**	**16QAM**	**64QAM**	**128QAM**	**256QAM**
MSE	3.8517×10^{-10}	3.8521×10^{-10}	3.8533×10^{-10}	3.8526×10^{-10}	3.8514×10^{-10}	3.8526×10^{-10}
MER [dB]	70.572	70.572	70.572	70.572	70.572	70.572

Table 3.4: MSE and MER for several different filter orders.

	Filter Order					
	16	**64**	**128**	**256**	**512**	**1024**
MSE	1.3317×10^{-3}	1.9853×10^{-8}	3.6452×10^{-9}	3.2091×10^{-9}	3.8513×10^{-10}	2.3038×10^{-10}
MER [dB]	5.1897	53.4555	60.8164	61.3699	70.5720	72.8092

BPSK, QPSK, etc. symbols used to modulate the OFDM subcarriers) and the demodulated data symbols at the receiver side after all the vPHY, hypervisor and channelizer processing. Here we also take the MER average over 10^5 iterations. The MER for the k-th vPHY is calculated as follows

$$\text{MER}_k = \mathbb{E}\left[10\log_{10}\left(\frac{\sum_{n=0}^{N_{\text{vPHY}}^{U}-1}(I_k^2 + Q_k^2)}{\sum_{k=0}^{N_{\text{vPHY}}^{U}-1}(I_k - \tilde{I}_k)^2 + (Q_k - \tilde{Q}_k)^2}\right)\right], \quad (3.14)$$

where N_{vPHY}^{U} represents the number of useful subcarriers (i.e., the subcarriers that are modulated with the data symbols), I_k is the In-phase value corresponding to the k-th reference symbol, Q_k is the Quadrature phase value corresponding to the k-th reference symbol, $\tilde{I}_k$ is the In-phase value corresponding to the k-th received symbol, and $\tilde{Q}_k$ is the Quadrature phase value corresponding to the k-th received symbol. The MER can be seen as a SNR measurements, where it calculates the distortion/interference caused by the multiplexing operation performed by the hypervisor.

The MER is used to verify the difference between the symbols of an ideal constellation, with points precisely at the ideal locations, and the received signal after all the vPHY, hypervisor, and channelizer processing. Therefore, basically, with MER, we want to check deviations from the ideal locations that the architecture/implementation can cause to the received signal.

In Table 3.3 we show the MSE and MER averages for several different modulation schemes. We employ the MER metric to verify how similar the system's output, after the channelizer, is to a *regular* single PHY, i.e., a PHY that does not have its signal multiplexed with other PHYs. We have not considered noise in this simulation. The MSE estimation values shown in the table were calculated averaging the error between PHY and vPHY OFDM symbols over 1×10^5 iterations. The MER averaging was also executed over the same number of iterations. For each iteration, the same set of randomly picked data bits modulates both the single PHY and the 12 vPHYs. The table shows that the MSE is quite low and almost the same for all modulation schemes and that the MER is high and the same for all

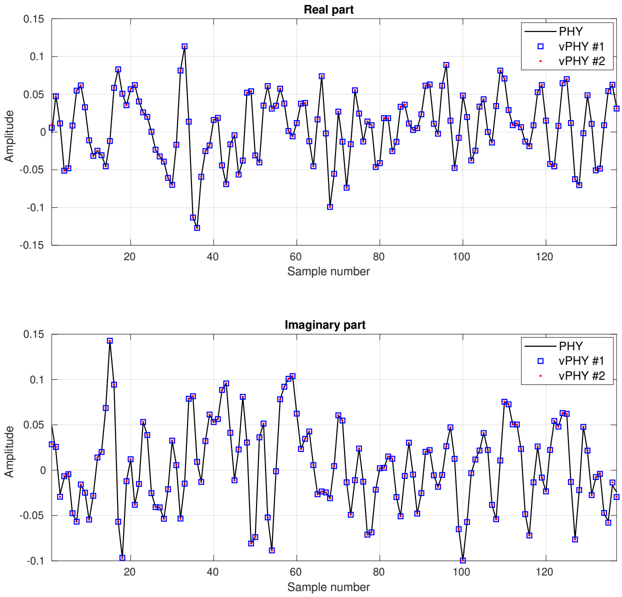


Figure 3.11: Comparison between OFDM symbols generated by a plain/single PHY and 2 *vPHYs.*

schemes, which means that both MSE and MER are independent of the employed modulation scheme. Additionally, it is worth mentioning that as all vPHYs are fed with the same set of bits, their resulting MSE and MER are the same.

In Table 3.4 we show the comparison of MSE and MER for several different filter order values. We have not considered the addition of noise in this simulation. The main idea here is to assess the signal's quality at the receiver side when we vary the channelizer's filter orders. The variations in MSE and MER are only the result of varying the filter order. As we noticed with the results in Table 3.3, the MSE and MER values are almost the same for all modulation schemes, and therefore, in this simulation, we use the same modulation scheme, QPSK, for all trials. As mentioned before, the filter order is important to guarantee a good signal fidelity. This is due to the fact that the higher the filter order, the closer it is to the perfect window filter frequency response, which does not impose any distortion to the filtered signal [57].

In Figure 3.11, depicts the comparison between OFDM symbols generated by a single (i.e., one that does not have its signal multiplexed) PHY and 2 vPHYs having their output signals multiplexed by the hypervisor. This simulation does not consider the addition of noise. Here, for the sake of comparison, the single PHY and the 2 vPHYs are fed with the same set of data bits. As can be seen, the vPHY OFDM symbols are quite similar to the OFDM symbol generated by the

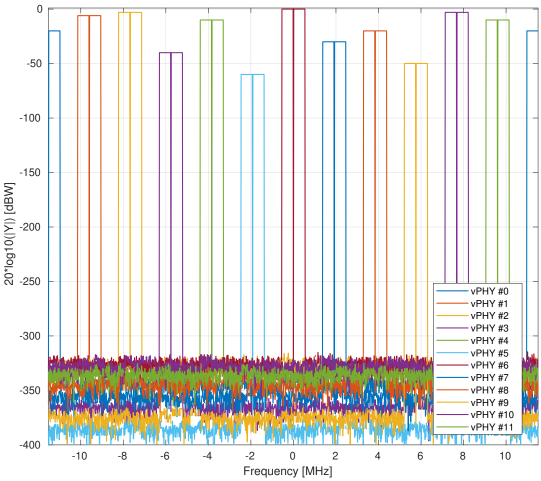


Figure 3.12: Frequency-domain representation of the wide baseband signal generated by the hypervisor.

single PHY. In theory, at the receiver side, after the channelizer, the 2 multiplexed vPHY signals must match the single PHY symbol since the multiplexing and channelization are expected to be transparent to the vPHY receivers. This simulation is meant to check if the multiplexing and demultiplexing (i.e., channelization) processing generates a *clean* signal, similar to one not multiplexed.

In Figure 3.12, we show the frequency-domain representation of the wide baseband signal generated by the proposed wireless hypervisor. This simulation does not consider the addition of noise. Here, in this figure, the hypervisor multiplexes, in the frequency domain, the signals of 12 vPHYs, M, where each vPHY uses 128 subcarriers, N_{vPHY}, totaling 1536 subcarriers, which is the number of points used by the IFFT block in the hypervisor, N. Each vPHY only modulates 72 subcarriers and leaves the remaining subcarriers along its edges as guard bands. The subcarrier at the center of each vPHY channel is set to 0, which is used to allow receivers to employ simpler/cheap direct-conversion (i.e., zero intermediate-frequency) RF front-end receivers. In the example, shown in the figure, 64QAM modulation is used to modulate the data signal. As can be seen, there are (i) a guard-band between consecutive vPHY signals, (ii) a null-subcarrier exactly at the center of each one of the vPHY transmissions, (iii) and different transmission levels for the vPHYs. As shown in the figure, it is possible to give independent gains to each vPHY, which is accomplished by multiplying the N_{vPHY}^{U} useful modulation symbols by a multiplication factor varying from 0 (no transmission power at all) to 1 (maximum transmission power used by the physical RF front-end). The noise floor in the figure is pretty low due to the precision used for the simulations, which

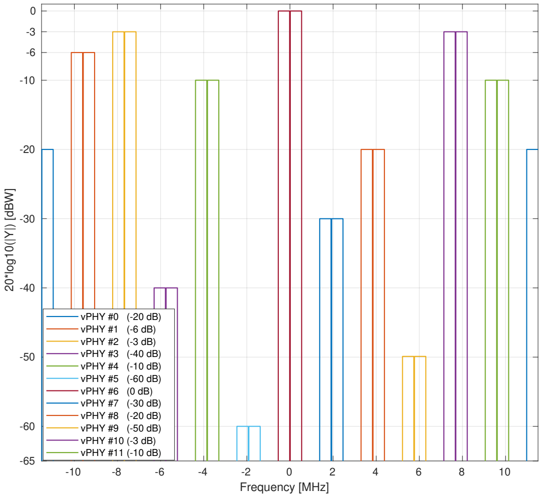
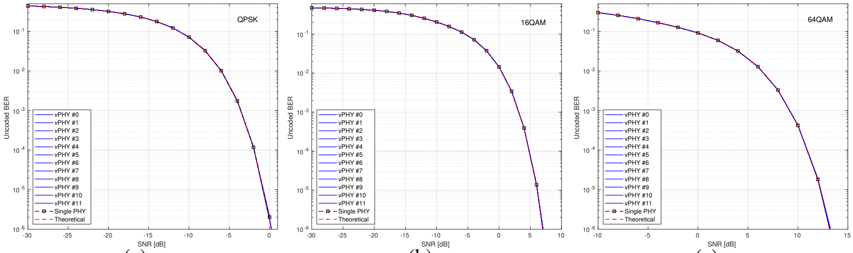


Figure 3.13: Closer look at the frequency-domain boost factor.

used double-precision floating-point variables. However, with a real world device such as an USRP, we will have fixed-point implementations. Such conversion from floating to fixed-point will increase the noise floor considerable.

Figure 3.13 gives a closer look at the boost factor, showing that it is possible to vary the transmitted power of each individual vPHY just by changing the factor used to multiple the useful subcarriers, N^{U}_{vPHY}. The legend on the figure shows the attenuation given to the default transmission power of a vPHY, which is around 0 [dBW] as shown by the vPHY centered around 0 [Hz].

Figure 3.14 presents uncoded Bit Error Rate (BER) results for a setup where 12 vPHYs concurrently have their signals multiplexed by the hypervisor and employ QPSK, 16QAM, and 64QAM modulation schemes. In each one of the sub-figures, we compare the BER for the 12 vPHYs against the BER achieved by a single PHY, i.e., there is no other signal being transmitted along with that of the PHY under test. In this simulation, we consider an AWGN channel and that all vPHYs have the same transmission power. The single PHY and each one of the 12 vPHYs modulate 72 (i.e., $\mathrm{N}^{U}_{\mathrm{vPHY}}$) out of 128 (i.e., $\mathrm{N}_{\mathrm{vPHY}}$) consecutive subcarriers by using QPSK, 16QAM, and 64QAM modulation schemes. The hypervisor's IFFT length, N, is set to 1536 and the cyclic prefix length, N_{CP}, is set to 108 samples. For each SNR point, 1×10^{6} iterations were run, where the total number of wrongly decoded bits and transmitted bits were calculated for the BER calculation. As can be seen, the vPHY BER curves exactly match the single PHY BER curve (dashed black curve with squares along it), meaning that there is no interference between the current transmissions. Additionally, it is also worth mentioning that all curves match the theoretical BER curves (red-dashed curve with dots along it), which can



(a) (b) (c)

Figure 3.14: BER curves for a setup where 12 *vPHYs concurrently have their signals multiplexed by the hypervisor and use: (**a**) QPSK, (**b**)* 16*QAM and (**c**)* 64*QAM modulation schemes.*

be approximated by (3.15) [58].

$$P_b \approx \frac{2(\sqrt{\gamma}-1)}{\sqrt{\gamma}\log_2(\gamma)}\text{erfc}\left(\sqrt{\frac{3\log_2(\gamma)(E_b/N_0)}{2(\gamma-1)}}\right), \tag{3.15}$$

where γ is the modulation order and E_b/N_0 is the bit energy over the power spectrum density. The results presented in Figure 3.14, are very important as they prove that the proposed architecture provides perfect isolation among all the signals being multiplexed. The perfect isolation is due to the orthogonality provided by the IFFT processing in the hypervisor, which guarantees that every single subcarrier, spaced of Δ_f [Hz], is mutually orthogonal to all other ones.

It is important to notice that there is a link between MSE and BER results. If the *reconstructed* signal, after the channelizer output, is not properly filtered, it will consequently affect the BER. That is, if the order of the polyphase filters are too low, the quality of the *reconstructed* will be poor as the signals from adjacent sub-bands will not be properly filtered, i.e., they will spill over the desired sub-band, *polluting* the desired signal. Therefore, the higher the filter order, the better is the signal quality.

Figure 3.15 shows the frequency-domain representation of the wide baseband signal generated by the hypervisor for 3 vPHYs with different but correlated numerologies. This simulation does not consider the addition of noise. These 3 vPHYs employ frequency spacings, Δ_f of 15, 7.5, and 3.75 [KHz], and CP lengths of 9, 18, and 36 samples, FFT/IFFT lengths of 128, 512, and 2048 points, and sampling rates of 1.92, 3.84, and 7.68 [MSps], respectively. In this simulation, we use a hypervisor with an IFFT of 6144 points with a sampling rate of 23.04 [MSps], which results in a subcarrier frequency spacing of 3.75 [KHz]. For all 3 vPHYs, we modulate 72 subcarriers and leave the others unused. As can be seen, even though the 3 vPHYs have only 72 useful subcarriers, we notice that due to the different frequency spacings, Δ_f, each one of them has a different occupied bandwidth.

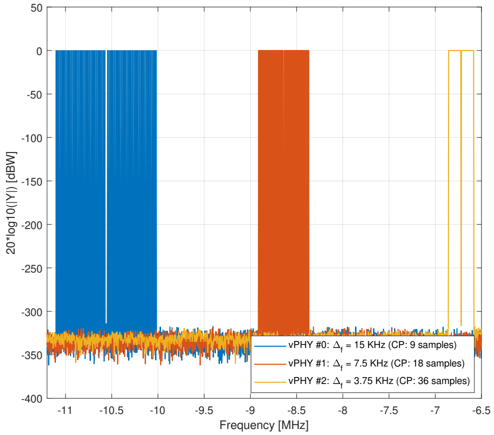
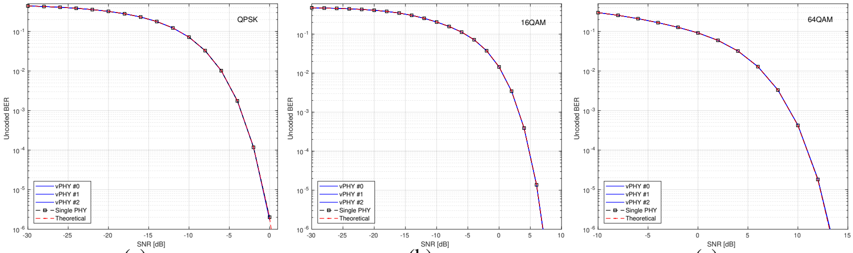


Figure 3.15: Frequency-domain representation of the wide baseband signal generated by the hypervisor for 3 *vPHYs with different but correlated numerologies.*

In the case of vPHY$\#0$, the useful subcarriers appear every 4 points so that the final subcarrier frequency spacing equals 15 [KHz]. In the case of vPHY$\#1$, the useful subcarriers are mapped into the Hypervisor's subcarriers every 2 points, resulting in a subcarrier frequency spacing of 7.5 [KHz]. In the case of vPHY$\#2$, the useful subcarriers are mapped consecutively into the Hypervisor's subcarriers, which results in a subcarrier frequency spacing of 3.75 [KHz]. The receiver side for receiving the signal of vPHY$\#0$ works as before, since we respect the subcarrier frequency spacing of 15 [KHz]. For receiving the signals of the other 2 vPHYs the receiver would need to use FFT with lengths of 512 and 2048 and assume subcarrier frequency spacing of 7.5 and 3.75 [KHz], respectively.

Figure 3.16 shows uncoded BER results for the setup used for plotting Figure 3.15 where the 3 vPHYs with different but correlated numerologies concurrently have their signals multiplexed by the hypervisor. These 3 vPHYs employ QPSK, 16QAM, and 64QAM modulation schemes for each one of the simulation results presented in the figure. In this simulation, we consider an AWGN channel and that all vPHYs have the same transmission power.

In each one of the sub-figures, we compare the achieved BER of the 3 vPHYs against the BER achieved by a single PHY in AWGN channel. The single PHY and each one of the 3 vPHYs modulate 72 subcarriers by using QPSK, 16QAM, and 64QAM modulation schemes. For each SNR point, 1×10^6 iterations were run, where the total number of wrongly decoded bits and transmitted bits were calculated for the BER calculation. As expected, the vPHY BER curves exactly match the single PHY BER curve, meaning that there is no interference between the current transmissions. Additionally, it is important to highlight that even though the vPHYs have different numerologies the Hypervisor is able to multiplex the signals



(a) (b) (c)

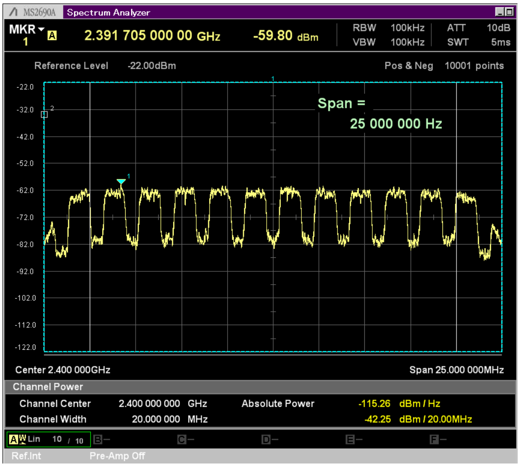
Figure 3.16: BER curves for a setup with 3 *vPHYs with different but correlated numerologies. Modulations used are (**a**) QPSK, (**b**)* 16*QAM and (**c**)* 64*QAM modulation schemes.*

so that the different receivers can successfully decode the signals. Moreover, it is also worth mentioning that all curves match the theoretical BER curves, which are approximated by (3.15). Again, as expected, the results presented in Figure 3.16, show that the proposed architecture provides perfect isolation among all the signals being multiplexed, even if they have different but correlated numerologies. This perfect isolation is due to the orthogonality provided by the IFFT processing in the hypervisor, which guarantees that every single subcarrier, spaced of $\Delta_f = 3.75$ [KHz], is mutually orthogonal to all other ones.

3.8 Experimental Evaluation

In this section, we present some experimental results carried out to validate and assess the functionality of the proof of concept implementation of the proposed architecture. All the experiments presented in this section were carried out with the prototype running on a desktop with an Intel Xeon E5-2650 v4 CPU (@2.2 GHz, 30 M cache, 9.60 GT/s QPI, Turbo, HT, 12 Cores/24 Threads, 105 Watts) with 128 GB of RAM memory connected to a x310 USRP with 10 Gigabit Ethernet link, and equipped with CBX-120 RF daughter-boards [59]. These RF daughter-boards operate from 1200 up to 6000 MHz with a bandwidth of 120 MHz [59].

For all the experiments presented in this section, each vPHY has a useful transmitting BW of 1.08 MHz, which is equivalent to 6 LTE RBs, and a guard-band of 420 KHz at each side of the transmitted spectrum, totaling 1.92 MHz of used BW (i.e., useful-band plus guard-band sections) per vPHY, totaling 12×1.92 MHz/vPHY $= 23.04$ MHz of occupied BW when we have 12 vPHYs operating concurrently. Due to the lack of full-blown and tested (i.e., operational) open-source projects that implement systems with different but correlated numerologies such as subcarrier frequency spacing, we will not be able to experiment with correlated numerologies. However, as shown in Section 3.7, working with different



MS2690A Spectrum Analyzer

MKR 1 A 2.391 705 000 00 GHz -59.80 dBm

RBW 100kHz ATT 10dB
VBW 100kHz SWT 5ms

Reference Level -22.00dBm

Pos & Neg 10001 points

Span = 25 000 000 Hz

Center 2.400 000GHz

Span 25.000 000MHz

Channel Power

Channel Center 2.400 000 000 GHz
Channel Width 20.000 000 MHz

Absolute Power -115.26 dBm / Hz
-42.25 dBm / 20.00MHz

A W Lin 10 / 10

Ref.Int

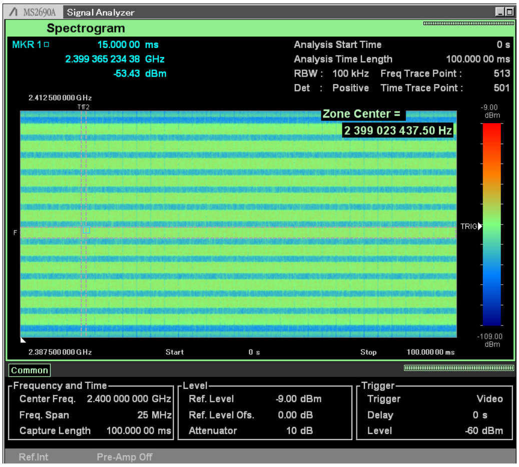
Pre-Amp Off

Figure 3.17: Spectrum of 12 *vPHYs concurrently transmitting at a center frequency of* 2.4 *GHz.*

but correlated numerologies would not be a problem to the proposed architecture.

Figure 3.17 shows the spectrum of 12 vPHYs transmitting concurrently. This figure was collected with an Anritsu MS2690A Signal Analyzer. The RF front-end center frequency was set to 2.4 GHz and Tx gain set to 3 [dB] with the USRP Tx output connected to the signal analyzer through a cable with 20 [dB] of attenuation. As can be seen, the total transmission BW spans over 23.04 MHz, which is equivalent to having 12 1.92 MHz-wide vPHYs with their center frequencies located at 1.92 MHz apart from each other. Additionally, we can also see that the two vPHYs transmitting at the right and left edges suffer from attenuation, which is caused by cascaded integrator-comb (CIC) filter roll-off. CIC filter are implemented in the USRP to provide decimation by an arbitrary programmable integer decimation factor; however, they present a very significant pass-band roll-off, which are often called *spectral droop* or *CIC roll-off* [60].

In Figure 3.17, we can also notice the Out-Of-Band (OOB) emissions inherent to OFDM modulation. It is possible to see that the *pollution* at the edges of each sub-band. In this specific case, the boost factor, rho_k, won't help with the OOB since it is due to the poor spectrum localization of OFDM modulation. Each sub-band can be seen as an independent OFDM-based PHY, so the pollution at adjacent channels is caused by the OOB emissions of each PHY. The multiplexed signal, the one with all the vPHY's signals, is already *polluted* and those OOB emissions get amplified by the RF amplifier. Possible solutions to mitigate the OOB problem are (i) apply windowing to the frequency domain sub-band signal before the IFFT and/or (ii) filter the individual sub-band signals, sum them up and send it to the USRP. This would be similar to Universal Filtered Multi-Carrier (UFMC)



MS2690A Signal Analyzer

Spectrogram

MKR 1 □ 15.000 00 ms
2.399 365 234 38 GHz
-53.43 dBm

Analysis Start Time 0 s
Analysis Time Length 100.000 00 ms
RBW : 100 kHz Freq Trace Point : 513
Det : Positive Time Trace Point : 501

Zone Center =
2 399 023 437.50 Hz

Start 0 s
Stop 100.000 00 ms

Common

Frequency and Time
Center Freq. 2.400 000 000 GHz
Freq. Span 25 MHz
Capture Length 100.000 00 ms

Level
Ref. Level -9.00 dBm
Ref. Level Ofs. 0.00 dB
Attenuator 10 dB

Trigger
Trigger Video
Delay 0 s
Level -60 dBm

Ref.Int Pre-Amp Off

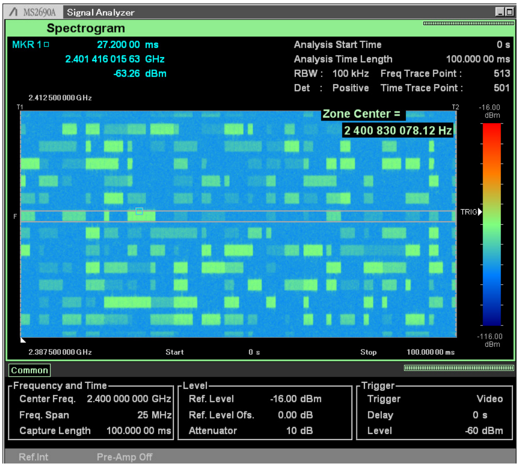
Figure 3.18: Spectrogram of 12 *vPHYs concurrently transmitting at a center frequency of* 2.4 *GHz.*

approach.

Figure 3.18 shows the spectrogram (time versus frequency) for the same experiment setup used to capture Figure 3.17. In this experiment all 12 vPHYs transmit in streaming mode, i.e., each vPHY transmits subframes all the time with no gap between subsequent subframes. The transmitted signal was captured for a period of 100 ms. It can be noticed that all 12 vPHYs transmit at the same time throughout the whole analysis interval, without any gap between consecutive subframes.

Figure 3.19 shows that the proof of concept prototype of the proposed hypervisor is able to handle discontinuous transmissions and to apply independent gains to each vPHY, as described in Section 3.3. Again, we used the same experiment setup used to capture Figure 3.17. In this experiment all 12 vPHYs transmit in discontinuous (burst) mode with random number of subframes transmitted in a row, channel number and frequency amplification factor. The number of subframes, channel number and frequency amplification factor of each vPHY are randomly selected between the ranges 0–11, 0–5, and 0–100% respectively. The transmitted signal was captured for a period of 100 ms. As can be seen, the prototype is also able to work on burst mode with independent frequency amplification factor for each vPHY. This result also shows that the prototype also supports runtime configuration of the number of transmitting vPHYs as we see that not all vPHYs might be transmitting during a period.

The following experiments are executed by adding a channel emulator between the Tx and Rx sides of the prototype. At the Tx side, the generated multiplexed signal, instead of being sent to the USRP HW is sent to an abstraction layer that emulates the HW and adds AWGN to the transmitted signal, next, the abstraction



```
MS2690A  Signal Analyzer
Spectrogram
MKR 1 □        27.200 00  ms        Analysis Start Time                    0 s
       2.401 416 015 63  GHz        Analysis Time Length         100.000 00 ms
                 -63.26  dBm        RBW :  100 kHz   Freq Trace Point :    513
                                    Det  :  Positive  Time Trace Point :   501
Zone Center =
2 400 830 078.12 Hz
2.387 500 000 GHz        Start    0 s              Stop   100.000 00 ms

Common
Frequency and Time                   Level                           Trigger
 Center Freq.   2.400 000 000 GHz     Ref. Level       -16.00 dBm     Trigger        Video
 Freq. Span                25 MHz     Ref. Level Ofs.    0.00 dB      Delay            0 s
 Capture Length   100.000 00 ms       Attenuator        10 dB         Level        -60 dBm
Ref.Int        Pre-Amp Off
```

Figure 3.19: Spectrogram of 12 *vPHYs with discontinuous transmissions and independent frequency gains at a center frequency of* 2.4 *GHz.*

layer transfers the noisy signal to the receiving side of the prototype. With these experiments, we only wanted to check the prototype's capacity to handle 12 concurrent vPHYs. As we use srsLTE's PHY implementation, we are not concerned about other impairments that PHY's baseband processing can treat.

Figure 3.20 shows the throughput measurements taken with the proposed architecture prototype for several MCS values and a duty cycle of 95.24 %. In this experiment, the prototype works in full-duplex mode (i.e., it is simultaneously transmitting and receiving) with 1 single vPHY (upper part of the figure) and 12 vPHYs (lower part of the figure) working at the same time. We adopt a full-duplex mode to check if this mode somehow impacts the measured throughout, as in full-duplex mode the prototype is being fully used. The measurements were taken with transmissions of 20 ms (i.e., 20 subframes) and a gap of 1 ms between subsequent transmissions, and therefore, a duty cycle of 95.24 %. The throughput is calculated as an average over 10 measurement intervals of 10 s each. During one measurement interval (i.e., 10 s) the total number of received bits from all vPHYs is counted and then divided by the interval to produce the throughput measured during that interval. As in this experiment we are only interested in the maximum throughput that can be achieved, the SNR on the link was set to 30 [dB] so that the packet reception rate for all MCS values was equal to 1. For the sake of comparison, the theoretical maximum throughput achieved by the *Streaming* mode (i.e., transmissions with a duty cycle equal to 100 %) is added to the figure. The theoretical maximum throughput is calculated by dividing the size in bits of an LTE transport block for each MCS value [61] by 1 ms. As can be seen, the measured prototype's throughput approaches the theoretical maximum throughput for

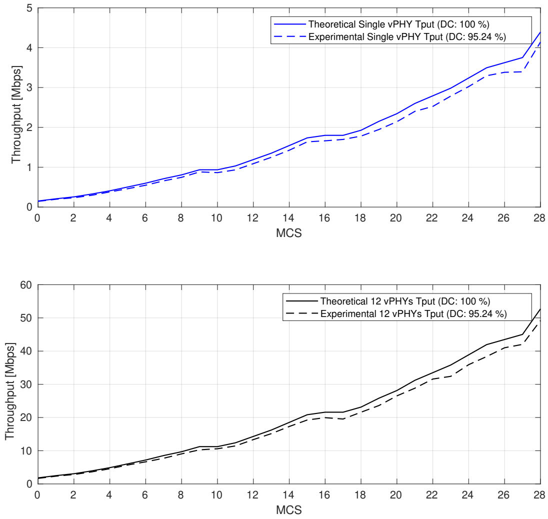
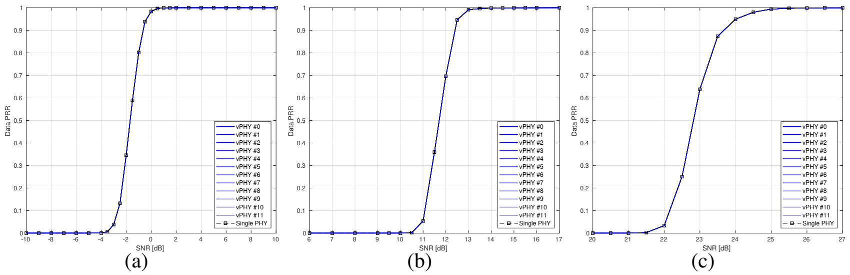


Figure 3.20: Prototype's throughput for single and 12 vPHYs over different MCS values.

all MCS values, yielding more than 4.14 Mbps in the single vPHY case and more than 49.2 Mbps in the 12 vPHYs case for MCS 28. Additionally, as can be also noticed, the operation in full-duplex mode has no visible impact on the achieved throughput. This is due to the powerful server, with 12 CPU cores, used to run the prototype.

Next, in Figure 3.21 we compare the data Packet Reception Ratio (PRR) of a single PHY against the data PRR of 12 vPHYs concurrently transmitting over a range of SNR values. In this experiment, 12 vPHYs concurrently have their signals multiplexed, transmitted and received by the prototype. The experiment uses 3 different MCS values so that all 3 modulation schemes employed by LTE standard (i.e., QPSK, 16QAM, and 64QAM) are tested. The PRR is calculated as the average over 10^5 Monte Carlo trials, where at each trial, the Tx side of the prototype sends either a single PHY signal or the multiplexed signal of 12 vPHYs. In each one of the sub-figures, we compare the PRR of the 12 vPHYs against the PRR obtained with a single PHY, i.e., there is no other signal being transmitted along with that of the PHY under test. As theorized earlier, the PRR curves of the 12 concurrent transmitting vPHYs match the PRR of the single PHY (i.e., the dashed black curve with squares along it). This means that there is no interference between the concurrent vPHY transmissions. These results prove that the prototype of the proposed architecture also, as shown before with the simulation results, provides isolation among all the signals being multiplexed by the prototype. The achieved isolation is due to the orthogonality provided by the IFFT processing implemented by the prototype. Therefore, as can be seen, the prototype behaves like



(a) (b) (c)

Figure 3.21: Data PRR curves for a setup with 12 *vPHYs with their signals multiplexed by the prototype and using: (**a**) MCS0 (QPSK), (**b**) MCS16 (*16*QAM), and (**c**) MCS28 (*64*QAM).*

12 single independent PHYs.

Figure 3.22 depicts the CPU and memory use of the prototype for several MCS values. These results compare CPU and memory use when the prototype must multiplex the signal of 1 and 12 vPHYs, respectively. The results in the figure were calculated by averaging CPU and memory usage values sampled every 200 ms during the duration of the experiment, which was set to 60 s. Each one of the vPHYs transmit 20 subframes in a row with a 1 ms gap between consecutive transmissions.

As can be observed, the CPU use increases as the MCS increases; however, there is no CPU starvation issue. The increase in CPU use is mainly due to the fact that as the MCS value increases (i.e., higher data rates), the turbo encoding (at Tx side), synchronization and turbo decoding (both at Rx side) processing tasks become more complex and consequently demand a lot more of CPU for data processing. For a MCS equal to 28 and 12 vPHYs concurrently transmitting, the CPU use is of approximately 450 %, meaning that the processing power of fewer than 5 cores is being employed, leaving the other cores in the idle state for large periods. On the other hand, we see that the memory use is practically constant for all MCS values and goes from around 0.5% to 2.2% for 1 and 12 vPHYs, respectively. Therefore, memory use is practically independent of the MCS value being used. This is an expected result as all memory being used by the prototype is pre-allocated during its initialization. Therefore, based on the results presented in Figure 3.22, it can be concluded that the prototype does not exhaust CPU or memory resources as the MCS value increases. These are important results, once they show that given the current server configuration, the prototype can be scaled to support even more vPHYs without exhausting CPU or memory resources.

Next, in Figure 3.23, we present the assessment of the CPU consumption of independent processing tasks making up the architecture prototype. For this assessment we employ the *callgrind* tool, which is part of the *valgrind* profiler. *Callgrind*

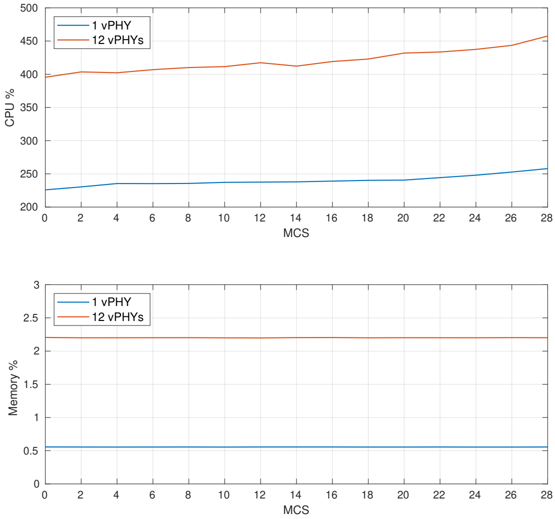


Figure 3.22: CPU and memory use of the architecture prototype for 1 *and* 12 *vPHYs.*

is a profiling tool that keeps the call history among functions in a program's run as a call-graph through the use of runtime instrumentation [17, 62]. The figure presents the functions with the highest CPU processing load (i.e., the most representative CPU consumers) for 3 different MCS values and the cases where 1 and 12 vPHYs are instantiated. The setup used for this experiment is the same as the one used during the experiment for CPU and memory profiling.

As can be noticed, channelization presents the highest CPU consumption throughout all test cases. Channelization is a quite heavy processing once it keeps always processing a bandwidth equivalent to the maximum number of configured vPHYs, which in this case is equal to 12, no matter the number of actually instantiated vPHYs. For 1 instantiated vPHY, inverse FFT is the second most consuming task; however, its CPU consumption remains constant for all considered MCS values as it does not depend on the MCS used. In the case that 12 vPHYs are instantiated, we see that memory copy, *memcpy*, increases its load from 15.35 % for MCS 0 to approximately 20 % for MCS 28. Compared to the 1 vPHY case, it is a drastic increase once it has a maximum CPU load of 7.41 % for MCS 28. For the 12 vPHYs case, the inverse FFT is the third most consuming task, ranging from 20.04 % for a MCS equal to 0 to 12.37 % for a MCS equal to 28. We see that in the second column, the one for 12 instantiated vPHYs, the IFFT processing load gradually decreases while *memcpy* gradually increases its CPU load. On the other hand, we also see that the CPU processing load of the memory set operation, *memset*, remains constant throughout the evaluated MCS values. It is also important to highlight that bit interleaving processing gets heavier as the MCS value increases,

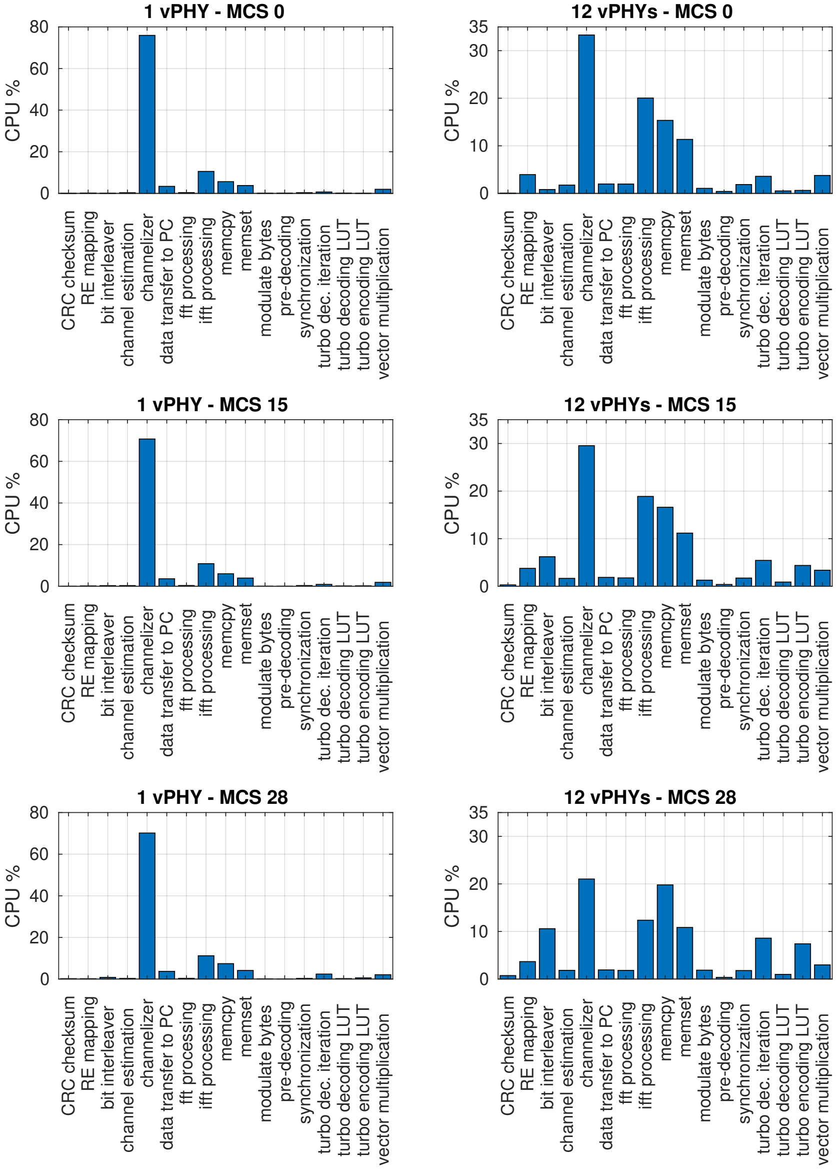


Figure 3.23: CPU profiling of individual components of the architecture prototype for 1 *and* 12 *vPHYs.*

consuming approximately 0.81 % of CPU time for a MCS equal to 0 and going to 10.59 % when the MCS value is made equal to 28.

3.9 Conclusions

In this chapter, we proposed a wireless spectrum hypervisor architecture that abstracts a RF front-end into a configurable number of virtual RF front ends. Our approach was proposed to improve spectral efficiency by efficiently using vacant gaps in congested spectrum bandwidths or employing network densification through infrastructure sharing. We provided a mathematical demonstration on how the proposed approach works and presented several simulation results proving its functionality and efficiency. Additionally, we presented an open-source and free proof of concept prototype of the proposed architecture and several experimental results validating its functionality and showing its performance.

Acknowledgment

This work was funded by the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 732174 – Orchestration and Reconfiguration Control Architecture (ORCA) project.

References

[1] Cisco, T. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper. Available online: http://url.cn/5maQcaS (accessed on 15 September 2019).

[2] IMT-2020 (5G) Promotion Group. 5G Vision and Requirements; Technical Report. Available online: http://www.imt-2020.org.cn/en/documents/download/3 (accessed on 2 December 2019)

[3] Hossain, E.; Hasan, M. 5G Cellular: Key Enabling Technologies and Research Challenges. *IEEE Trans. Instrum. Meas.* 2015, 18, 11–21.

[4] Bjornson, E.; Hoydis, J.; Sanguinetti, L. Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency. *Found. Trends Signal Process.* **2017**, *11*, 154–655, doi:10.1561/2000000093.

[5] Lopez-Perez, D.; Ding, M.; Claussen, H.; Jafari, A.H. Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments. *IEEE Commun. Surveys Tuts.* 2015, *17*, 2078–2101.

[6] Huo, Y.; Lu, F.; Wu, F.; Dong, X. Multi-Beam Multi-Stream Communications for 5G and beyond Mobile User Equipment and UAV Proof of Concept Designs. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.

[7] de Figueiredo, F.A.P.; Dias, C.F.; de Lima, E.R.; Fraidenraich, G. Capacity Bounds for Dense Massive MIMO in a Line-of-Sight Propagation Environment. *Sensors* 2020, *20*, doi:10.3390/s20020520

[8] Huo, Y.; Dong, X.; Xu, W.; Yuen, M. Enabling Multi-Functional 5G and Beyond User Equipment: A Survey and Tutorial. *IEEE Access* 2019, *7*, 116975–117008.

[9] Bhushan, N.; Li, J.; Malladi, D.; Gilmore, R.; Brenner, D.; Damnjanovic, A.; Sukhavasi, R.T.; Patel, C.; Geirhofer, S. Network Densification: The Dominant Theme for Wireless Evolution into 5G. *IEEE Commun. Mag.* **2014**, *52*, 82–89.

[10] Aboba, B. Virtual Access Points. Available online: http://www.drizzle.com/aboba/IEEE/11-03-154r1-I-Virtual- Access-Points.doc(accessed on 16 September 2019).

[11] Kumbhkar, R.; Sridharan, G.; Mandayam, N.B.; Seskar, I.; Kompella, S. Design and implementation of an underlay control channel for NC-OFDM-based networks. In Proceedings of the Conference on Information Science and Systems (CISS), Princeton, NJ, USA, 16–18 March 2016.

[12] 5G; NR; Physical Channels and Modulation (3GPP TS 38.211 version 15.2.0 Release 15). Available online: http://url.cn/5DBQNyx (accessed on 16 September 2019).

[13] Kaltenberger, F.; de Souza, G.; Knopp, R.; Wang, H. The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap. In Proceedings of the Workshop on Smart Antennas (WSA), Vienna, Austria, 24–26 April 2019.

[14] de Figueiredo, F.A.P. *Wireless-Spectrum-Hypervisor*. GitHub Repository. Available online: Https://github.com/zz4fap/Wireless-Spectrum-Hypervisor (accessed on 15 July 2019).

[15] Akyildiz, I.F.; Wang, P.; Lina, S.-C. SoftAir: A software-defined networking architecture for 5G wireless systems. *J. Comput. Networks* 2015, *85*, 1–18.

[16] Liang, C.; Yu, R. Wireless network virtualization: A survey, some research issues and challenges. *IEEE Commun. Surv. Tutorials* 2014, *17*, 358–380.

[17] de Figueiredo, F.A.P.; Jiao, X.; Liu, W.; Mennes, R.; Jabandzic, I.; Moerman, I. A spectrum sharing framework for intelligent next generation wireless networks. *IEEE Access* 2018, *6*, 60704–60735.

[18] Harris, F.J.; Dick, C.; Rice, M. Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications. *IEEE Trans. Microw. Theory Tech.* 2003, *51*, 1395–1412.

[19] Litwin, L. FIR and IIR digital filters. *IEEE Potentials* 2000, *19*, 28–31.

[20] Such, G.; Schraml, M.; Knopp, A. Frequency Domain Channelizer: A computationally efficient method for non-uniform channelization. In Proceedings of the GNU Radio Conference, Henderson, NV, USA, 17–21 September 2018

[21] Mendes, J.; Jiao, X.; Garcia-Saavedra, A.; Huici, F.; Moerman, I. Cellular Access Multi-Tenancy through Small Cell Virtualization and Common RF Front-End Sharing. In Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH), Singapore, 9–12 October 2017.

[22] Kist, M.; Rochol, J.; DaSilva, L.A.; Bonato, C. HyDRA: A hypervisor for software-defined radios to enable radio virtualization in mobile networks. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Atlanta, GA, USA, 1–4 May 2017.

[23] Kist, M.; Rochol, J.; DaSilva, L.A.; Bonato, C. SDR Virtualization in Future Mobile Networks: Enabling Multi-Programmable Air-Interfaces. In Proceedings of the IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018.

[24] FCC. *Report of the Spectrum Efficiency Working Group*; Federal Communications Commission, Report; Available Online: Https://transition.fcc.gov/sptf/files/SEWGFinalReport_1.pdf (accessed on 20 November 2002).

[25] Baig, G.; Alistarh, D.; Karagiannis, T.; Radunovic, B.; Balkwill, M.; Qiu, L. Towards unlicensed cellular networks in TV white spaces. In Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT'17), Incheon, Korea, 12–15 December 2017.

[26] Maloku, H.; Limani Fazliu, Z.; Ibrani, M. A Survey on Coexistence in Heterogeneous Wireless Networks in TV White Spaces. *Wirel. Commun. Mob. Comput.* 2018, *2018*, doi: 10.1155/2018/7256835

[27] Tan, K.; Shen, H.; Zhang, J.; Zhang, Y. Enable flexible spectrum access with spectrum virtualization. In Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Bellevue, WA, USA, 16–19 October 2012; pp. 47–58.

[28] de Figueiredo, F.A.P.; Aniceto, N.F.T.; Seki, J.; Moerman, I.; Fraidenraich, G. Comparing f-OFDM and OFDM Performance for MIMO Systems Considering a 5G Scenario. In Proceedings of the 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 30 September–2 October 2019.

[29] Amendment of the Commissions Rules with Regard to Commercial Operations in the 3550–3650 MHz Band. Available online: https://www.fcc.gov/document/commercial-operations-3550-3650-mhz-band (accessed on 16 September 2019).

[30] Sohul, M.M.; Yao, M.; Yang, T.; Reed, J.H. Spectrum access system for the citizen broadband radio service. *IEEE Commun. Mag.* 2015, *53*, 18–25.

[31] Mennes, R.; Claeys, M.; De Figueiredo, F.A.P.; Jabandzic, I.; Moerman, I.; Latre, S. Deep Learning-Based Spectrum Prediction Collision Avoidance for Hybrid Wireless Environments. *IEEE Access* 2019, *7*, 45818–45830.

[32] Camelo, M.; Shahid, A.; Fontaine, J.; De Figueiredo, F.A.P.; De Poorter, E.; Moerman, I.; Latre, S. A Semi-Supervised Learning Approach Towards Automatic Wireless Technology Recognition. In Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Newark, NJ, USA, 11–14 November 2019.

[33] DARPA. The World's First Collaborative Machine-Intelligence Competition to Overcome Spectrum Scarcity. Available online: https://www.spectrumcollaborationchallenge.com/ (accessed on 7 August 2016).

[34] de Figueiredo, F.A.P.; Stojadinovic, D.; Maddala, P.; Mennes, R.; Jabandzic, I.; Jiao, X.; Moerman, I. SCATTER PHY: An Open Source Physical Layer for the DARPA Spectrum Collaboration Challenge. *MPDI Electron.* 2019, *8*, doi:10.3390/electronics8111343

[35] Kar, P.; Dappuri, B. Site Survey and Radio Frequency Planning for the Deployment of Next Generation WLAN. In Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2018.

[36] Flexi Multiradio BTS RF Module and Remote Radio Head Description- Doc. Num. DN0951745. Available online: http://url.cn/5KttbDe (accessed on 16 September 2019)

[37] RRH-LTE-2600 Wildcat - Remote Radio Head- LTE; Product Specification. Available online: http://url.cn/5WskaE4 (accessed on 17 September 2019)

[38] Nutaq. TitanMIMO-X Technology, Product Specification. Available online: https://www.nutaq.com/products/titanmimo/titanmimo-x/technology (accessed on 21 June 2019).

[39] Samdanis, K.; Costa-Perez, X.; Sciancalepore, V. From Network Sharing to Multi-tenancy: The 5G Network Slice Broker. *IEEE Commun. Mag.* 2016, *54*, 32–39.

[40] 3GPP TR 28.801 – Telecommunication management;Study on management and orchestration of network slicing for next generation network, Available online: https://itectec.com/archive/3gpp-specification-tr-28-801/ (accessed on 17 September 2019)

[41] Xia, L.; Kumar, S.; Yang, X.; Gopalakrishnan, P.; Liu, Y.; Schoenberg, S.; Guo, X. Virtual Wi-Fi: Bring Virtualization from Wired to Wireless. In Proceedings of the ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, Newport Beach, CA, USA, 9–11 March 2011.

[42] Bhanage, G.; Vete, D.; Seskar, I. SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), Miami, FL, USA, 6–10 Decembber 2010.

[43] Santos, J.F.; Kist, M.; van de Belt, J.; Rochol, J.; DaSilva, L.A. Towards Enabling RAN as a Service - The Extensible Virtualisation Layer. In Proceedings of the IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019.

[44] Kokku, R.; Mahindra, R.; Zhang, H.; Rangarajan, S. NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks. *IEEE/ACM Trans. Netw.* 2012, *20*, 1333–1346.

[45] Foukas, X.; Marina, M.K.; Kontovasilis, K. Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture. In Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom), Snowbird, UT, USA, 16–20 October 2017; pp. 127–140.

[46] Sachs, J.; Baucke, S. Virtual radio: A framework for configurable radio networks. In Proceedings of the International Conference on Wireless Internet, Maui, HI, USA, 17–19 November 2008; pp. 1–7.

[47] Zaki, Y.; Zhao, L.; Goerg, C.; Timm-Giel, A. LTE wireless virtualization and spectrum management. In Proceedings of the Wireless and Mobile Networking Conference (WMNC), Budapest, Hungary, 13–15 October 2010; pp. 1–6.

[48] Frigo, M.; Johnson, S.G. The Design and Implementation of FFTW3. *Proc. IEEE*, 2005, *93*, pp. 216–231.

[49] Gomez-Miguelez, I.; Garcia-Saavedra, A.; Sutton, P.D.; Serrano, P.; Cano, C.; Leith, D.J. srsLTE: An Open-Source Platform for LTE Evolution and Experimentation, Available online: https://dl.acm.org/doi/abs/10.1145/2980159.2980163 (accessed on 17 September 2019).

[50] Ettus Research LLC. USRP X Series. Available online: Https://www.ettus.com/product/category/USRP-X-Series (accessed on 10 November 2019).

[51] National Instruments. *Overview of the NI USRP RIO Software-Defined Radio*. White Paper, December 2015. Available online: Http://www.ni.com/white-paper/52119/en/ (accessed on 10 November 2019).

[52] Ettus Research LLC. *USRP Hardware Driver*. Available online: Http://files.ettus.com/manual/ (accessed on 10 November 2019).

[53] Ettus Research LLC. *Products*. Available online: Https://www.ettus.com/product (accessed on 10 November 2019).

[54] Hintjens, P. *ZeroMQ Messaging for Many Applications*; O'Reilly Media: Sebastopol, CA, USA, 2013.

[55] Google. Protobuf - Protocol Buffers - Google's Data Interchange Format. Available online: Http://code.google.com/p/protobuf/ (accessed on 12 April 2019).

[56] Nikaein, N.; Marina, M.K.; Manickam, S.; Dawson, A.; Knopp, R.; Bonnet, C. OpenAirInterface: A Flexible Platform for 5G Research. *SIGCOMM Comput. Commun. Rev.* 2014, *44*, 33–38.

[57] Abdoli, J.; Jia, M.; Ma, J. Filtered OFDM: A new waveform for future wireless systems. In Proceedings of the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Stockholm, Sweden, 28 June–1 July 2015.

[58] Cho, K.; Yoon, D. On the general BER expression of one- and two-dimensional amplitude modulations. *IEEE Trans. Commun.* 2002, *50*, 1074–1080.

[59] Ettus Research LLC. CBX 1200-6000 MHz Rx/Tx (120 MHz, X Series only). Available online: Https://www.ettus.com/product/details/CBX120 (accessed on 21 September 2019).

[60] Hogenauer, E.B. An economical class of digital filters for decimation and interpolation. *IEEE Trans. Acoust. Speech Signal Process.* 1981, *29*, 155–162.

[61] 3GPP.LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures (3GPP TS 36.213 Version 13.0.0 Release 13), Available online: http://url.cn/5iqyAnD (accessed on 21 December 2019)

[62] Weidendorfer, J.; Kowarschik, M.; Trinitis, C. A Tool Suite for Simulation Based Analysis of Memory Access Behavior. In Proceedings of the 4th International Conference on Computational Science (ICCS), Krakow, Poland, 6–9 June 2004.

4
A Spectrum Sharing Framework for Intelligent Next Generation Wireless Networks

In the previous chapters, we presented and described efficient SDR architectures and techniques for radio HW virtualization and softwarization. From this chapter onward, we propose advanced SDR architectures and techniques for efficient and fair spectrum sharing. Therefore, in this chapter, we introduce a novel open-source SDR-based spectrum sharing framework that can be employed to devise disruptive techniques to optimise the sub-optimal use of today's radio spectrum.

⋆ ⋆ ⋆

This chapter is based on the homonymous article by

Felipe A. P. de Figueiredo, Xianjun Jiao, Wei Liu, Ruben Mennes, Irfan Jabandzic, and Ingrid Moerman.

Published in IEEE Access, vol. 6, pp. 60704-60735, Oct. 2018.

Abstract The explosive emergence of wireless technologies and standards, covering licensed and unlicensed spectrum bands has triggered the appearance of a huge amount of wireless technologies, with many of them competing for the same spectrum band instead of harmoniously sharing it. Unfortunately, the wireless spec-

trum is a scarce resource, and the available frequency bands will not scale with the foreseen demand for new capacity. Certain parts of the spectrum, in particular the license-free ISM bands, are overcrowded, while other parts, mostly licensed bands, may be significantly underutilized. As such, there is a need to introduce more advanced techniques to access and share the wireless medium, either to improve the coordination within a given band, or to explore the possibilities of intelligently using unused spectrum in underutilized (e.g., licensed) bands. Therefore, in this chapter, we present an open source SDR-based framework that can be employed to devise disruptive techniques to optimize the sub-optimal use of radio spectrum that exists today. We describe three use cases where the framework can be employed along with intelligent algorithms to achieve improved spectrum utilization. Additionally, we provide several experimental results showing the performance and effectiveness of the proposed framework.

4.1 Introduction

The demand for wireless broadband services is insatiable. Current wireless networks can not offer spectrum BW and network capacity to meet the growing traffic demands (e.g., consumers, enterprises, vertical industries, service providers, etc.) forecast for the coming years. According to [1], wireless networks will see an increase in usage by 47 $\%$ CAGR by 2021, reaching staggering 49 Exabytes per month, when for example, speeds are expected to reach peaks of 10 Gbps. Additionally, devices ranging from smartphones to wearable fitness recorders to smart kitchen appliances are voraciously competing for BW. With everything considered, around 50 billion wireless devices are expected to be competing for access to wireless communications networks in the next coming years. By 2030, the demand for wireless access is foreseen to be 250 times what it is today [1].

Next generation wireless networks are expected to provide broadband access wherever needed and also support a diversified range of services including everything from self-driving cars to virtual reality, robotic surgery and IoT [2, 3]. Connections in the order of one to dozen of Gbps to vehicles, high speed trains, data-intensive services (e.g., augmented reality, immersive 360° experiences, etc.) are some of the applications that will drive the demand for larger coverage area, lower latency and higher capacity at reduced cost in next generation wireless networks [4]. Unfortunately, the wireless radio spectrum is a scarce resource, and the available wireless BW does not scale with the needed wireless BW [5]. Hence, as can be seen, the development of technologies that can tackle the ever increasing traffic and service demands while combating the imminent scarcity of spectrum BW is of vital importance for next generation wireless networks.

The development of such technologies will be driven by three broad use cases families, namely, Enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC). They aim at improving performance, scalability and (cost/energy) efficiency of the current wireless networks such as LTE. Therefore, these use cases and their direct requirements will demand huge improvements in comparison with the previous generation of IMT systems [6]. Next, we briefly discuss each one of them and list some possible applications [6, 7].

- **eMBB**: this use case focus on enhancements to the data rate, user density, latency, capacity and coverage of the actual wireless broadband access networks [8]. Some of its applications are: (a) High-speed mobile broadband; (b) Augmented Reality; (c) Virtual Reality; (d) Smart office environment, i.e. all devices wirelessly connected; (e) Pervasive video, e.g., high resolution video communications, ultra high definition multimedia streaming, etc.

- **URLLC**: this use case enables devices and machines to establish communication links with ultra-reliability, high availability and very low latency, which makes it ideal for the following applications [9]: (a) Wireless Industrial Control; (b) Factory Automation; (c) Remote Surgery; (d) Cellular Vehicle-to-Anything (C-V2X) communications; (e) Drone communications; (f) Smart Grids; (g) Public Safety.

- **mMTC**: this use case focus on enabling communications between devices that are low-cost, massive in number and battery-driven [7, 8]. It is intended to support applications like: (a) Smart metering; (b) Smart Cities; (c) Asset tracking; (d) Remote monitoring, e.g., field and body sensors.

In consequence of these new use cases, the next generation of wireless networks will be required to deliver ultra-fast speeds, very low latency, huge coverage area and excellent reliability to dozen of billions of wireless devices. It is worth mentioning that not all of these requirements have to be met at the same time, as different services have different subsets of QoS requirements. Additionally, different services with different QoS constraints further have to share the same spectrum band.

Some approaches that can be used to cope with the expected traffic increase are: (i) increased/improved spectrum availability (such as introduction of new spectrum bands or more efficient/intelligent use of the available ones) [10]; (ii) introduction of technical enhancements (such as new radio interfaces, codecs, use of multicast transmissions (i.e., simultaneous transmission of data to a group of users through point-to-multipoint communication), reduction of energy consumption, etc.); (iii) new network structures and topologies aiming at increased network density (i.e., the addition of more radio sites to the same geographical area and the consequent shrinkage of cell sizes) [11]; (iv) traffic offloading to less occupied spectrum bands (offloading traffic onto unlicensed bands makes capacity available for other users in the license band and improves user experience for devices being served in the unlicensed band. It could also happen within unlicensed bands, e.g., offloading traffic from 2.4 GHz to 5 GHz band) [12]. As we discuss in the next section, our research focus on making more efficient and intelligent use/sharing of the available spectrum bands.

4.1.1 Motivation

Most of today's channel allocations separate wireless systems by splitting the spectrum into fixed and exclusively licensed bands that are assigned over large and geographically defined regions. This approach restricts access to the spectrum in exchange for guaranteed interference-free communications. These allocations of spectrum are human-driven and not adaptive to the dynamics of traffic demand and supply. At any given time, many allocated spectrum bands are unused by their licensees while other bands are completely flooded. For example, a report from the FCC's Spectrum Policy Task Force (SPTF) shows that 85 % of current allocated radio frequency bands are either partially or completely unused at different times across geographical areas [13]. This kind of channel allocation scheme tremendously wastes the spectrum capacity and creates unnecessary scarcity [5].

Spectrum sharing, where more than one user shares the spectrum band, either in time and/or space, is one possible and highly viable approach to achieve better spectrum utilization (i.e., combat spectrum BW scarcity) and meet the foreseen increase in traffic demand. Additionally, spectrum sharing can be categorized into two different types: (i) sharing in unlicensed bands; and (ii) sharing in licensed bands. Type (i) can be further split into spectrum sharing in unlicensed bands (i-A) with and (i-B) without an anchor in licensed bands. Some technologies using sharing type (i-A) are LTE-LAA and LTE-Unlicensed (LTE-U). LTE-LAA and LTE-U occupies licensed and unlicensed bands at the same time through the use of carrier aggregation. Signaling and traffic with specific QoS requirements will use licensed bands, while less critical traffic is offloaded onto unlicensed bands [12, 14, 15]. Some technologies employing sharing type (i-B) are Wi-Fi and MulteFire, which is a LTE-LAA based technology that works solely in unlicensed bands without the need for an anchor in licensed band [16]. For Type (i) sharing, simple sense-and-avoid techniques such as LBT and Carrier-Sensing Adaptive Transmission (CSAT) are applied in order to comply with regulations [10]. The need for LBT is mandatory in some regions (e.g., Europe and Japan), and not mandatory in other regions (e.g., US and China) [12]. For Type (ii) sharing, the techniques employed are Spectrum Access Systems (SAS) in the US and Licensed Shared Access (LSA) in Europe. These techniques will be deployed to allow unused spectrum bands of incumbents to be shared with licensees. Such networks will not use LBT schemes, but will rely on a central licensing authority to ensure that interference is avoided by setting exclusion/protection zones. Access to shared, licensed-shared and unlicensed spectrum bands will be of huge importance to next generation wireless networks as they have the potential to provide significant gains, both in terms of spectrum efficiency and spectrum reuse. These gains translate into higher data speeds for the users, enables new capabilities such as prioritized guaranteed resources (i.e., guaranteed time/frequency resources that provide each network with QoS similar to that it would get from licensed spectrum use) and new deployment scenarios [17].

Technologies devised for spectrum sharing in unlicensed bands, such as LTE-U, LTE-LAA and MulteFire, avoid interfering with neighbor networks by employing simple sense-and-avoid techniques [14], however, in the foreseen dense and diverse future use of spectrum, these simple schemes will not survive, because only local information about the medium availability (which brings about an issue known in the literature as the *hidden node problem* [14]) and naive avoidance mechanisms are insufficient to maintain end-to-end QoS of a myriad of competing wireless links. On the other hand, technologies for spectrum sharing in licensed bands, such as SAS and LSA, avoid interference by having a central arbitrator to decide how the licensees will access the incumbents' band. However, in order to ensure reliable QoS to incumbents and licensees, these technologies can not depend exclusively on geo-location and spectrum data bases to decide how they grant access the spectrum [18]. These licensed-sharing technologies will require algorithms that provide fair and demand-driven allocation of power and spectrum resources to licensees so that both incumbents and licensees have their QoS requirements met.

Consequently, in order to reap the full capacity out of all types of available spectrum bands (i.e., licensed, unlicensed or licensed-shared bands) and attain stable and adequate communications links, it is clear from the discussion above that technologies for next generation wireless networks have to go beyond simple interference avoidance, frequency isolation (i.e., fixed channel allocation schemes) and geo-location/data base based access approaches. Radios being devised for next generation networks will have to employ greater intelligence and collaboration to avoid interference while maximizing spectrum usage and capacity. These radios will need to intelligently collaborate with their peers in order to manage and optimize the use of the available spectrum without prior knowledge of each other's operating characteristics [19]. Therefore, collaborative intelligent radios should be capable of collaborating with other previously unknown radios and adapt their communication schemes to optimize the combined wireless spectrum usage across all radios. These radios have to be devised not only to communicate reliably in congested and contested environments but also to share the available spectrum band without any central coordination or spectrum pre-allocation planning across

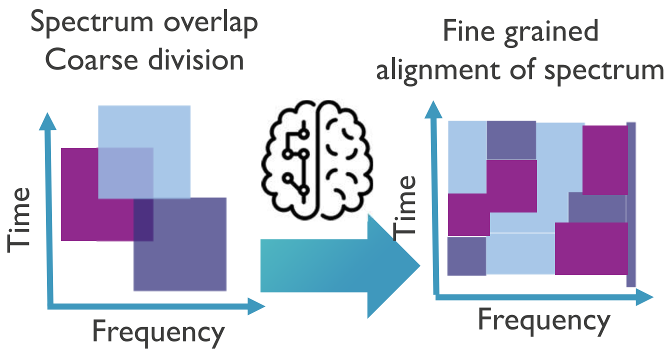


Figure 4.1: Collaborative intelligent spectrum sharing.

a wide range of heterogeneous radios.

In summary, next generation wireless networks will be required to add advanced intelligent algorithms to their radios so that they can collectively develop strategies that optimize the use of the wireless spectrum in ways not possible with today's intrinsically simple and often centralized approaches. As a consequence, researchers need to rethink the current strategies adopted for spectrum access and come up with new wireless paradigms where radio networks can autonomously collaborate, understand the current state of the available spectrum, and jointly reason how to share it more efficiently, i.e., jointly exploit opportunities to attain optimum/efficient use of the available spectrum BW. Figure 4.1 depicts one possible outcome of the intelligent collaboration among radios with different needs and characteristics. By applying an intelligent collaborative spectrum sharing scheme, the radios can move from a coarse and often overlapping spectrum distribution (i.e., poor spectrum usage due to the absence of cooperation among radios) to a more fine-grained and aligned spectrum allocation (i.e., optimized spectrum usage).

Intelligent and collaborative spectrum sharing models will drive the research and development efforts towards novel concepts aiming at increasing the spectrum usage efficiency and enhancing the coexistence among heterogeneous wireless networks [20]. These concepts will cover a wide range of topics, including the application of ML and AI algorithms to the sharing problem, new deployment models, dynamic spectrum allocation and regulatory aspects, novel radio access technologies, standalone operation of networks (i.e., operation without mobile infrastructure support), new network architectures, ecosystems and business models, etc.

In recent years, prototyping and experimental validation of innovative wireless technologies have gained importance due to the ever increasing complexity of the wireless ecosystem in emerging next generation wireless scenarios. Such complex systems cannot be analyzed anymore with theoretical models or in simulators without applying oversimplified mathematical models and assumptions, far away from limitations imposed by real hardware and real-life environments. Testbeds play a major role in developing and testing new wireless communications technologies and systems and, as with any disruptive technology, prototyping using realistic testbeds is the best way to truly understand the performance trade-offs and limitations. Based on that, several research initiatives are proposing the design of flexible, re-configurable and re-programmable prototyping frameworks and platforms for evaluating, comparing and validating the performance trade-offs of innovative wireless devices, communication techniques, network models, services, etc. in realistic testbed environments.

SDRs [21, 22] are radio communication systems where several transceiver components, which are typically implemented in hardware (e.g., filters, modulators, demodulators, etc.), are instead, implemented in software on a personal computer or embedded system. SDR platforms provide flexibility in reconfiguration of baseband algorithms, software and reprogramming of RF parameters. Moreover, the concept of SDR is very encouraging for the development of novel wireless communications technologies, as software programming allows much faster development cycles and real-life experiments to be conducted, for example, at local or remote testbeds.

In order to ensure that the next generation of wireless networks can get the most out of collaborative and intelligent spectrum sharing techniques in real-world environments, we present in this chapter an open source and runtime configurable SDR-based framework, which is suitable for researching and prototyping novel spectrum sharing and coexistence mechanisms in realistic environments [23, 24]. The framework is composed of three main modules, namely, slot-based physical PHY layer, spectrum sensing and collision avoidance (i.e., listen before talk) modules. The proposed framework is implemented using UHD software API [25] and runs on commercially available off-the-shelf (COTS) hardware devices such as the USRP [26], which is a well-known platform for SDR development.

The main contributions of this chapter are summarized next.

- Proposal of an open-source SDR-based framework for intelligent spectrum sharing research in the context of next generation wireless networks. The proposed framework can be used to prototype and experimentally test intelligent algorithms devised to dynamically optimize the use of radio spectrum. We provide detailed explanation on each one of the modules making up the framework. The framework source code can be downloaded from its GitHub project page [23].

- Description of three use cases where the proposed framework can be used to dynamically optimize the use of radio spectrum, combating spectrum

Table 4.1: Comparison of the related framework solutions.

Framework	Open Source	Real-time configurability through control messages	RF Monitor	LBT	Discontinuous Tx	Cost
OAI [27]	Yes	No	No	No	No	-
openLTE [28]	Yes	No	No	No	No	-
srsLTE [29]	Yes	No	No	No	No	-
CONTACT [33]	No	Not informed	No	No	Yes	-
Amarisoft [30]	No	No	No	No	No	4500 to 8000
NI's LTE Framework [31]	No	No	No	No	No	> $66000

scarcity. The use cases show how the framework can be used to leverage intelligent spectrum sharing research for next generation wireless networks.

- Extensive experimental results showing the efficiency of the proposed framework. The results include experiments with f-OFDM, closer spectrum coexistence, collision avoidance and measurements of the throughput, CPU and memory utilization.

This chapter is organized as follows. In section 4.2, related works are compared with and distinguished from our work. Section 4.3 describes the proposed spectrum sharing framework for intelligent next generation wireless networks. Section 4.4 presents in detail a highly flexible slot-based PHY that can be used in collaborative intelligent spectrum sharing research. Section 4.5 describes the RF monitor module and the spectrum sensing algorithm used by this module to assess the medium state. Section 4.6 introduces a totally configurable software based LBT module. Section 4.7 presents some use cases with suggestions on how the proposed framework could be employed in intelligent next generation wireless networks. In section 4.8, we present and discuss the results of several experiments performed with the proposed framework. Finally, section 4.9 presents our conclusions and indicates directions for future work.

4.2 Related Work

Three of the most well-known open source LTE frameworks are Eurecom's OAI [27], openLTE [28] and Software Radio Systems' srsLTE [29]. OAI is compliant with LTE release 8.6 and implements only a subset of release 10. Additionally, it only supports 5, 10, and 20 MHz BW and the code structure is complex and difficult to customize. OpenLTE's source code is well organized and can be customized to some extent, however, it lacks detailed documentation, e.g., there is no information on compliance with any 3GPP release and it has a very silent mailing list. Furthermore, it is still incomplete and with several features unstable or under development. On the other hand, srsLTE's source code is well organized with a modular structure, has good documentation, a very active mailing list and can be easily customized. However, at the time of performing this work, it was only compliant with LTE Release 8 and implemented only a few features of Release 9. Moreover, these open source frameworks can only be configured through configuration files (i.e., static configuration) and none of them offers any mechanism or feature for spectrum sharing.

The Amarisoft LTE-100 platform is a commercial and closed source SDR-based LTE network suite. This solution is compliant with 3GPP LTE Release 14 [30], however, as far as we are concerned, it does not support any of the features necessary for the operation in unlicensed bands, which are the base for spectrum sharing approaches. This platform can not be configured in real-time, only allowing file based configuration. The cost of the Amarisoft LTE-100 software suite ranges from 4500 to 8000 depending on licensing type (fixed or floating) and number of channels. Moreover, as it is a closed-source solution, it can not be modified.

On the other hand there is the National Instruments (NI)'s LabVIEW Communications LTE-Advanced Application Framework [31]. This proprietary framework implements a subset of the 3GPP LTE Release 10 and provides support for the FlexRIO PXIe-7975/7976R and USRP RIO devices. The LTE framework is easy to be modified, mainly due to LabView's graphical programming language, allows real-time prototyping and is extensively validated but has no built in coexistence feature. In [32], NI customized the LTE framework in order to implement some LTE-U and LTE-LAA features like discontinuous transmission and LBT. However, the customized framework can not be bought separately from the whole LabVIEW 802.11 and LTE-Advanced (LTE-A) Application Frameworks, once it is sold as a Real-time LTE/Wi-Fi Coexistence Testbed. Additionally, the customized LTE framework can only be configured through a GUI and only allows threshold and Contention Window (CW) size parameters to be configured in real-time through the GUI. Moreover, this is a quite expensive solution, costing more than $66000.

A framework, named CONTACT, to explore emerging coexistence techniques among multiple RAT in both wireless communication and computer networking is presented in [33]. The proposed framework is divided into three layers: radio access, network and control layers. However, the framework only implements a simple preemptive sharing scheme at the radio access layer, i.e., it implements CSAT to sense channel usage and adjust the on and off LTE cycling based on Wi-Fi usage [34]. Such schemes do not take collaborative and intelligent spectrum sharing into account and can not be deployed world-wide as some countries require the use of contention based access mechanisms [34]. The framework only allows real-time configuration of the ON/OFF duty cycle, however, the authors do not mention by which means (e.g., through pre-defined control messages over network) it is carried out. Additionally, the proposed framework is not publicly available either as open or closed source.

To the best of the authors' knowledge there is no open source framework that offers researchers and practitioners the flexibility and the necessary modules (i.e., LBT and RF monitor) to devise, implement and experimentally test novel spectrum sharing schemes. Additionally, none of the above mentioned solutions allows users to easily plug-in other modules, receive information and change parameters in real-time through pre-defined messages over the network. Table 4.1 summarizes the comparison of characteristics presented by each one of the related solutions.

Next, we list some related simulation works that can make use of the proposed framework to also validate the ideas through experimental validation.

In [35] the authors propose the use of particle swarm optimization to learn the behavior of spectrum usage, helping to allocate spectrum dynamically. They also introduce a intelligent and optimal relay selection algorithm for effective selection of relay nodes, which enhances amplify and forward relay selection algorithm using intelligent agents.

A spectrum allocation solution using multi-agent system cooperation that enables secondary users to utilize the amount of available spectrum, dynamically and cooperatively is proposed in [36]. The agents are deployed on primary and secondary users that cooperate to achieve a better spectrum usage.

A Q-Learning based dynamic duty cycle selection mechanism is proposed in [37] for the configuration of LTE transmission gaps, so that a fair coexistence, i.e., spectrum sharing, between LTE and Wi-Fi networks is guaranteed. Simulation results show that the proposed Q-Learning based approach improves the overall system capacity performance by 19 % and Wi-Fi capacity performance by 77 % when compared to a scenario with fixed duty cycles where highest aggregate capacity is achieved. The results show that the approach enables effective coexistence of LTE and Wi-Fi systems in the unlicensed spectrum bands.

In [38] the authors demonstrate that a Neural Network (NN) can accurately predict slots in a Multiple-Frequency Time Division Multiple Access (MF-TDMA) network. Through spectrum observation, the proposed Neural Network models are able to do online learning and predict the behavior of spectrum usage a second in advance. Results show that the proposed approach reduces the number of collisions by half when nodes follow a Poisson traffic distribution and a reduction by a factor of 15 when more periodic traffic patterns are used.

In [39] the authors propose a NN based approach that adapts LBT's CW size based on the predicted number of Negative Acknowledgments (NACKs) for all subframes in a Channel Occupancy Time (COT) of LTE-LAA. The correct configuration of the CW size is of utmost importance to avoid collisions or to resolve contention among colliding radios. The proposed approach learns from previous experiences how many NACKs per subframe of a COT were received under certain channel conditions. After the learning phase, it is able to predict the number of NACKs for all subframes in a COT without having to wait (at least 4 ms) for

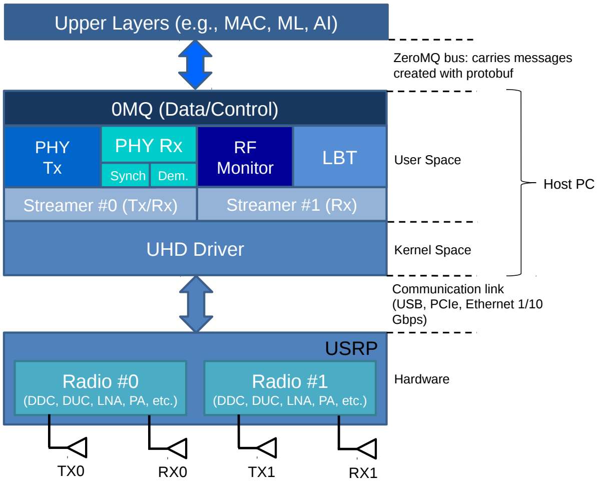

Upper Layers (e.g., MAC, ML, AI)
ZeroMQ bus: carries messages created with protobuf
0MQ (Data/Control)
PHY Tx
PHY Rx
Synch
Dem.
RF Monitor
LBT
User Space
Host PC
Streamer #0 (Tx/Rx)
Streamer #1 (Rx)
UHD Driver
Kernel Space
Communication link (USB, PCIe, Ethernet 1/10 Gbps)
USRP
Radio #0 (DDC, DUC, LNA, PA, etc.)
Radio #1 (DDC, DUC, LNA, PA, etc.)
Hardware
TX0
RX0
TX1
RX1

Figure 4.2: Proposed Framework Architecture.

delayed HARQ feedbacks. The CW size is exponentially increased upon the reception of a NACK for each subframe of a COT. Results show that the proposed approach provides better trade-off between fairness to Wi-Fi and LTE-LAA in terms of throughput and latency when compared to state-of-the-art approaches.

All the above mentioned works propose novel ideas to improve the spectrum utilization, however, all the results are simulation based. These ideas need to be evaluated to check their limits and capabilities of achieving improved spectrum utilization with real-world equipment. Therefore, the aim of our work is to fill this gap by providing a framework that can be used to implement these ideas, assess and validate them trough experimental validation.

4.3 Proposed Framework

In this section the proposed framework is described in details. It is comprised of three main modules: (i) slot-based PHY layer (for discontinuous data transmission and reception), (ii) RF Monitor (for spectrum sensing purposes) and (iii) LBT (for RAT coexistence). The high level architecture of the proposed framework is shown in Figure 4.2. All the main modules are connected to the ZeroMQ (Data/Control) module, also known as 0MQ, which interconnects the framework with upper layers through the ZeroMQ bus [40]. This module manages the exchange of control and statistics messages between the framework and upper layers. The framework is used in conjunction with Ettus USRP X family of SDR devices including NI's RIO platforms [41, 42] and communicates with it through the UHD driver and its APIs [26].

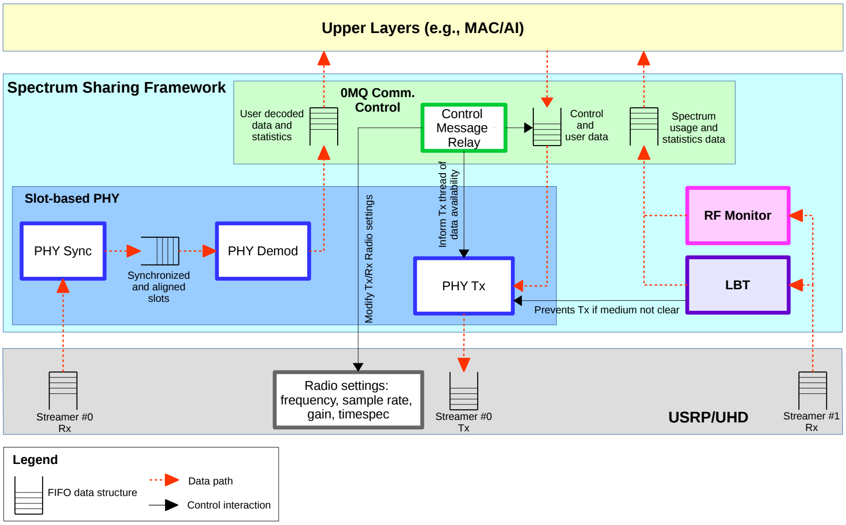

Upper Layers (e.g., MAC/AI)
Spectrum Sharing Framework
0MQ Comm. Control
User decoded data and statistics
Control Message Relay
Control and user data
Spectrum usage and statistics data
Slot-based PHY
PHY Sync
Synchronized and aligned slots
PHY Demod
Modify Tx/Rx Radio settings
Inform Tx thread of data availability
PHY Tx
RF Monitor
LBT
Prevents Tx if medium not clear
Streamer #0 Rx
Radio settings: frequency, sample rate, gain, timespec
Streamer #0 Tx
USRP/UHD
Streamer #1 Rx
Legend
FIFO data structure
Data path
Control interaction

Figure 4.3: Threading architecture of the proposed spectrum sharing framework.

USRP devices are hardware platforms developed for SDR, and are commonly used by research labs, universities, and hobbyists. USRP devices are controlled through the UHD driver. The driver provides an API that enables controlling and accessing all features provided by USRP devices, while abstracting the low level implementation details of the hardware. The most important operations provided by the UHD driver fall into two classes, namely, control and streaming. Control operations are used to set frequency, sample rate, gain, etc. of the radios. Streaming operations are carried out through streamer objects, which allow applications running on the host to connect to Tx and Rx sample streamers offered by the USRP device. These samples come from the radios available in the devices. An Rx streamer object allows applications to receive samples from the device and a Tx streamer object allows applications to transmit samples to the device [26].

The slot-based PHY exclusively uses Streamer #0, both for Tx and Rx sample streams, and is fed by Radio #0. The RF Monitor and LBT modules share Streamer #1 (both of them only use the Rx sample stream), which is fed by Radio #1. Streamers #0 and #1 can be configured independently, i.e., different TX/RX frequencies, gains and samples rates can be set for each of them independently. For performance and priority management reasons, each one of the modules (ZeroMQ Data/Control processing, PHY Tx, PHY Rx Synchronization, PHY Rx Data Demodulation, RF Monitor and LBT) runs on an exclusive thread. The reason why PHY and the sensing modules (i.e., RF Monitor and LBT) use different and independent Streamers is due to the fact that we want to decouple their operations. The RF Monitor and LBT modules use an exclusive streamer (i.e., Streamer #1) so that they can independently (from PHY RF parameters) and uninterruptedly (i.e., it is not desired to time-share a Streamer between PHY and the sensing modules) assess any configurable spectrum BW at any desired center frequency and Rx gain. With this approach, both PHY and the sensing modules will always be receiving at their independently configured center frequencies, BWs and Rx gains without being affected by or affecting each other's operation. In the current implementation, only one of the two sensing modules, i.e., either RF Monitor or LBT, is started and connected to Streamer #1. The module to be started is configured through command line during initialization of the framework.

Communication with the proposed framework is entirely realized through a well-defined interface designed with Google's Protocol Buffers (protobuf) [43] for data serialization coupled with the ZeroMQ messaging library [40] for distributed exchange of control, statistics and data messages. Implementing the ZeroMQ push-pull pattern allows local or remote upper layers' real-time configuration of several parameters and reading of several pieces of information/statistics provided by the framework. Based on the ZeroMQ logic, all modules are able to exchange control and data messages following a non-blocking communication paradigm. The proposed framework was designed to be totally decoupled and independent of upper layers modules (i.e., MAC, intelligence, etc.), not posing any constraints on hardware and software adopted by them. This way, upper layer modules can employ any type of hardware (e.g., GPUs, FPGA, etc.) or software algorithms in their implementation. Additionally, by adopting protobuf and ZeroMQ, the proposed framework offers other modules/layers the flexibility to use any kind of programming language implementation, making it easier to integrate multiple independent modules into a single working system. The flexible communication interface provided by the framework makes it ideal for deployment in testbeds and allows different upper layers to be plugged in, hence, an important enabler for intelligent spectrum sharing experimentation towards next generation of wireless networks.

Figure 4.3 illustrates the different layers composing the proposed spectrum sharing framework and the threads within each one of them. Red dashed arrows indicate data paths while black arrows indicate control/information interaction between threads.

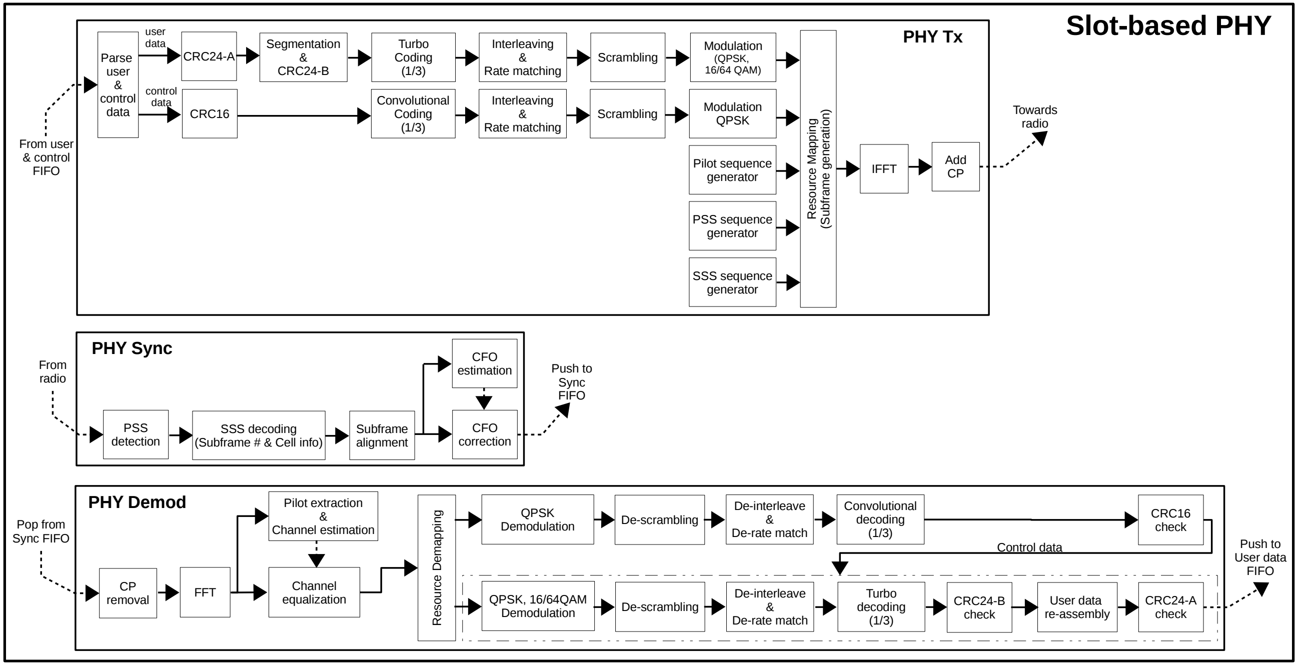


Figure 4.4: Block diagram of the slot-based PHY transmitter and receiver sides.

Through the use of the proposed framework, a myriad of real-world experiments can be performed, ranging from research of new physical layer techniques (e.g., higher order modulation schemes, novel waveforms and numerologies, channel coders, etc.) to RAT coexistence in unlicensed and licensed-shared spectrum bands (e.g., comparison of the fairness between different contention-based mechanisms such as CSAT and LBT, impact of different parameters such as COT, idle time, Tx power, PHY BW, spectrum overlap, MCS on the performance of Wi-Fi and LTE networks, etc.) to collaborative and intelligent radio networks (i.e., networks where AI and ML components act as the brain of the system and control various aspects of medium access and PHY layers to ensure adaptability, while also running some sort of collaboration protocol [44, 45]).

In the next sections, we describe each one of the main modules present in the proposed framework.

4.4 Slot-based Physical Layer

In spectrum sharing scenarios, several networks might share the medium at the same time and therefore, continuous access to it may not be possible all the time. With discontinuous transmissions, it is possible to have a better use of the available spectrum band and to coordinate its usage with other networks in an opportunistic/intelligent/collaborative way. Additionally, continuous access is the preferred approach for systems operating in exclusive licensed bands where there is no need for sharing, and is not really suited to fair medium sharing between co-located networks. Moreover, in unlicensed spectrum, some regulatory authorities do not allow continuous transmissions and limit the maximum duration of a transmission burst [46].

A PHY layer supporting a discontinuous transmission feature would have a great value to spectrum sharing research, as it could be used to leverage techniques devised to achieve fairer coexistence and higher spectral efficiency in scenarios where time-frequency resources are shared. Hence, as part of our framework for spectrum sharing research, we decided to develop a discontinuous transmission-based PHY, which transmits data bursts in small transport units called time slots, or slots for short. A slot is the container through which data is exchanged in the network.

We have based the development of the proposed slot-based PHY on the LTE PHY standard, as it offers several advanced features including high spectrum efficiency, multiple BWs, high peak data rates, mobility, multi-user access, flexible time framing and time-frequency structure, link adaptation with adaptive modulation and coding schemes and multiple-input and multiple-output (MIMO) [47, 48, 50]. Moreover, the next generation of wireless networks (i.e., 5G) will naturally evolve from LTE -based standards [47], i.e., they will be made available through improvements in LTE, LTE-A and LTE-A Pro technologies [51]. Consequently, building upon the LTE PHY standard makes it easier to further extend the slot-based PHY to be compliant with 5G standards and further evolutions. The proposed slot-based PHY is built upon the srsLTE library [29], and therefore, absorbs and evolves on top of the existing LTE features. srsLTE is a free and open-source LTE software library developed by Software Radio Systems (SRS) [29].

In order to stay aligned with both LTE standards and 5G initiatives [50, 51], we decided to adopt an OFDM based waveform. OFDM is a mature technology, which is vastly implemented in a great number of products due to its several advantages such as robustness to severe multipath fading, low implementation complexity, easy integration with MIMO, simple channel estimation, etc. [50]. Additionally, OFDM allows for enhancements such as waveform windowing/filtering, which can effectively minimize out-of-band spurious emissions [52]. It is also worth mentioning, that compared to other popular OFDM-based technologies like Wi-Fi [53] and Digital Video Broadcasting-Terrestrial (DVB-T) [54], LTE was devised, since its inception, for multi-user communications by assigning subsets of 12 subcarriers, also known as a physical resource block (PRB) or RB for short, over multiple 1 ms long subframes to individual users. This flavor of multiple access is known as Orthogonal OFDMA and gives LTE a very flexible way of allocating time/frequency resources to multiple concurrent users.

As mentioned earlier, the proposed slot-based PHY module is split into three submodules, namely, PHY Tx, PHY Rx Synchronization and PHY Rx Demodulation where each one of them runs on an exclusive, standalone thread. The reason for having a multi-threaded PHY implementation is that it allows independent critical and/or time-consuming tasks to be executed simultaneously (i.e., concurrently). Computing performance and efficiency is improved by taking advantage of concurrency. Allied with multi-core enabled CPUs, the multi-threaded PHY naturally supports full-duplex communications mode, i.e., one PHY can simultaneously transmit and receive at different frequencies, which consequently results in a higher throughput. PHY Tx thread is responsible for modulation and transmission of data (i.e., user and control data). PHY Rx Synchronization thread is responsible for the Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS) detection, CFO estimation/correction and slot (subframe) time-alignment tasks. CFO is estimated based on the PSS signal [55]. PHY Rx Demodulation thread takes care of user and control data demodulation, i.e, OFDM demodulation (FFT processing and CP removal), channel estimation/equalization, resource demapping, Physical Downlink Control Channel (PDCCH) decoding and Physical Downlink Shared Channel (PDSCH) decoding. The slot-based PHY receives data and control messages from the ZeroMQ Data/Control module. Decoded user data and statics related to the PHY operation (Rx/Tx statistics) are sent directly to the upper layers through 0MQ bus. Figure 4.4 depicts block diagrams of the slot-based PHY transmitter (PHY Tx thread) and receiver sides (PHY Sync and PHY Demod threads).

Regarding numerology, the proposed PHY supports LTE numerology with the same subcarrier spacing (15 kHz) and the CP lengths (normal and extended CPs). Different CP lengths are used to accommodate different levels of inter-symbol interferences (ISIs) at different frequencies, coverage and mobility scenarios [51]. However, it also supports the configuration of different (larger and smaller) subcar-

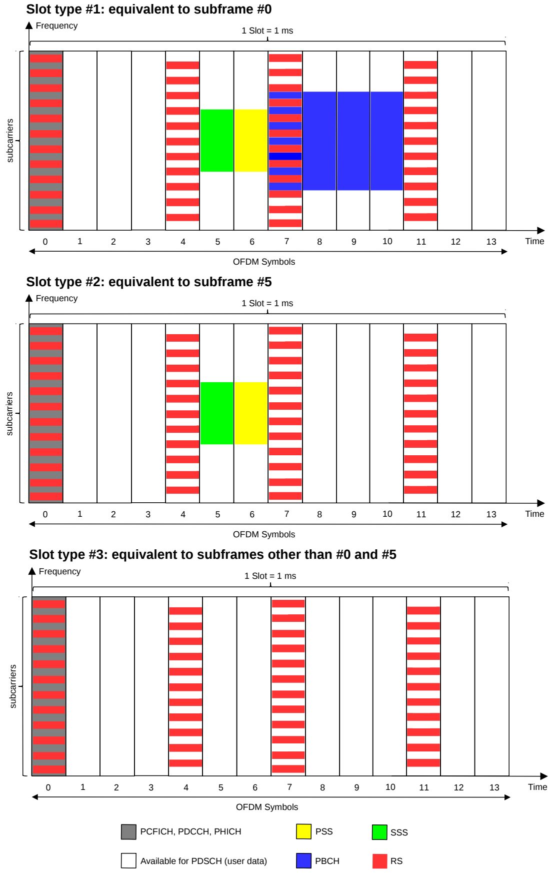


Figure 4.5: Proposed slot types for normal CP length.

rier spacings and CP lengths by adopting different sampling rates. Larger subcarrier spacing can be used to (i) decrease latency due to resulting smaller subframe durations, which is very important for URLLC applications requiring very low latencies, (ii) mitigate inter-carrier interference (ICI) in Vehicle to Everything (V2x) applications at high frequency bands (due to Doppler shift caused by high-speed scenarios) [48]. On the other hand, smaller subcarrier spacing is interesting for (i) extended coverage, due to increased Power Spectral Density (PSD), and (ii) high delay spread scenarios (due to longer CP) at low frequency bands [50, 51].

All LTE predefined BWs (1.4, 3, 5, 10, 15 and 20 MHz) are supported. PHY channel BW can be changed through command line at start up, or in real-time, through the Tx control message, which is introduced later in this section.

Differently from the LTE PHY downlink, which is continuously transmitting signals, the proposed slot-based PHY employs discontinuous (i.e., bursty) transmission of slots. A (time)-slot is the basic transmission unit of the proposed PHY and each one is 1 ms long. Depending on its type, a slot might carry synchronization signals, reference signals, broadcast, control and user data.

Three different types of slots, following the same definition as the LTE subframes, are proposed [47, 50]. All the three slot types for the normal CP length are depicted in Figure 4.5. The first type, which is equivalent to subframe #0 in the LTE standard, carries synchronization signals (PSS/SSS), reference signals, broadcast, user and control data. The second type is equivalent to LTE's subframe #5 and carries synchronization signals (PSS/SSS), reference signals, user and control data. The third type, which is equivalent to LTE's subframes other than #0 and #5 (i.e., #1 to #4 or #6 to #9), only carries reference signals, user and control data. It is important to notice that the RBs intended to carry user data can be split into different sections and allocated to different concurrent users. All slot types carry user data, signals for channel estimation/equalization, and control data, which holds information on how to decode each one of the possible user data sections. The first two slot types are used for transmission detection, as they carry synchronization signals. The first slot type is used whenever there is a need to transmit broadcast information. As it does not carry synchronization signals, the third slot type can only be used if it is transmitted after one of the other two types. Similarly to an LTE subframe, each slot can have 14 or 12 OFDM symbols depending on the configured CP length, i.e., normal or extended CP [47]. Moreover, each one of the three slot types is self-decodable, i.e., they always carry control data necessary for the receiver to correctly decode the user data section. The control data part of the slot is used at the PHY receiver side to automatically detect the number of allocated RBs, the location of the allocated RBs in the resource grid, and the MCS used to transmit data of a specific user. By following this approach, upper layers do not need to specify the number of allocated RBs, RB location and MCS before every slot reception. Upon correct user data reception, PHY informs to the upper layers the number of received bytes and the corresponding MCS.

The proposed PHY allows bursty transmissions with variable COT, i.e., the number of slots to be transmitted in a row without any gap (i.e., idle time) between them is variable. The number of slots in a COT, i.e., frame, is derived based on MCS, number of RBs and data length (i.e., number of bits to be transmitted) parameters sent by upper layers in the control message. The minimum COT is equal to 1 ms and is equivalent to a slot. Variable COT enables the support of different traffic loads and channel occupancy. Every slot can carry a pre-defined number of bits, which is based on the MCS and number of RBs used for a transmission [50].

The proposed slot-based PHY can operate with two different data transmission modes. The first mode can be used to establish communications with multiple nodes in ad-hoc or mesh networks, depending on routing capabilities implemented in the upper layers connected to the PHY (i.e., routing at layer 3 of the OSI model). This mode is a simplification of the LTE channel's structure and uses the PDSCH for user data transmissions in both directions, i.e., downlink and uplink, in order to take advantage of both multi-user communications and high performance Forward Error Correction (FEC) provided by turbo codes [50]. Additionally, this mode also uses Physical Physical Control Format Indicator Channel (PCFICH) and PDCCH

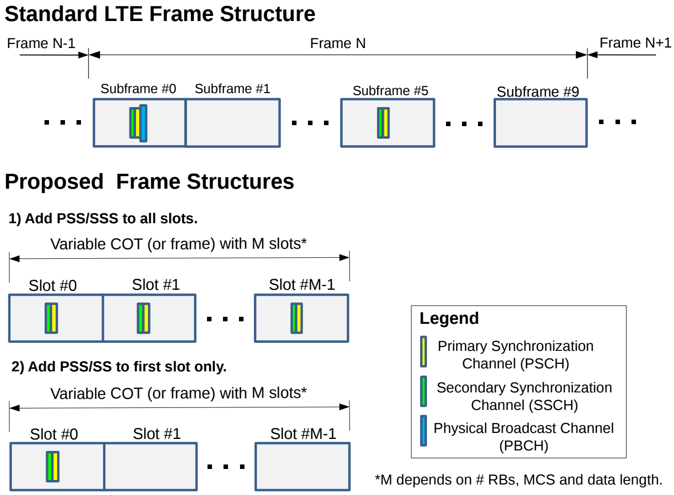

Standard LTE Frame Structure
Frame N-1
Frame N
Frame N+1
Subframe #0
Subframe #1
Subframe #5
Subframe #9
Proposed Frame Structures
1) Add PSS/SSS to all slots.
Variable COT (or frame) with M slots*
Slot #0
Slot #1
Slot #M-1
2) Add PSS/SS to first slot only.
Variable COT (or frame) with M slots*
Slot #0
Slot #1
Slot #M-1
Legend
Primary Synchronization Channel (PSCH)
Secondary Synchronization Channel (SSCH)
Physical Broadcast Channel (PBCH)
*M depends on # RBs, MCS and data length.

Figure 4.6: Standard LTE versus Proposed Frame Structures.

for control purposes and Cell-Specific Reference Signal (CRS) for channel estimation and equalization. The second mode operates exactly with the same channel's structure as adopted in the LTE standard, i.e., it has all downlink and uplink physical layer channels as defined in the standard. This mode can be used to establish communications the same way as in a cellular mobile network. In both modes, OFDMA can be used to establish communications with multiple users at the same time.

Regarding frame structures, we propose two different types, as depicted in Figure 4.6. In the first one, PSS and SSS sequences are added to all slots in a COT, allowing for better time-frequency tracking at the receiver side. This frame structure allows the receiver to easily perform slot synchronization every 1 ms without the necessity of complex tracking algorithms. A downside of this structure type is that PSS and SSS detection is time-consuming and consequently, decreases the maximum achievable throughput as the period of time required to detect and decode slots increases. In the second proposed frame structure, PSS and SSS signals are added only to the very first slot of a COT. This frame structure improves the throughput as PHY only needs to synchronize once (i.e., detect PSS, decode SSS, time-align and estimate/correct CFO), consequently, decreasing the overall slot decoding time [56]. However, this frame structure requires a more complex tracking/synchronization algorithm and the necessity to add information about the number of slots in a COT to the first slot, which means, if the first slot is lost, none of the subsequent slots can be decoded. Moreover, if the CFO is wrongly estimated or changes throughout the duration of a COT (i.e., frame), then the slots might be incorrectly decoded.

The first proposed frame structure is more robust to time and frequency fluctuations and should be used in cases where transmitter and receiver nodes are moving

Table 4.2: slot-based PHY real-time configurable parameters and statistics.

Message	Parameter	Type	Unit	Range
Tx control	Number of RBs	uint32	-	1-100
	MCS	uint8	-	0-28
	Tx gain	uint32	dB	depends on HW [1]
	Tx channel	uint32	-	≥ 0
	Tx PHY BW	uint8	MHz	0-6 [2]
	Transmission timestamp	uint64	s	≥ 0
	Frame structure type	uint8	-	0/1 [3]
	Data length	uint32	-	> 0
	User data	uchar[]	-	uchar range
Rx control	Rx channel	uint32	-	≥ 0
	Rx gain	uint32	dB	depends on HW [1]
	Rx PHY BW	uint8	MHz	0-6 [2]
Rx statistics	CQI	uint8	-	0-15
	RSSI	float	dBW	float range
	Noise	float	dBW	float range
	Decoded MCS	uint8	-	0-28
	Slot error counter	uint32	-	≥ 0
	Decoding time	uint32	ms	≥ 0
	Data length	uint32	-	≥ 0
	Received data	uchar[]	-	uchar range
Tx statistics	Coding time	uint32	ms	≥ 0
	Number of transmitted slots	uint32	-	≥ 0

as it provides better synchronization and fewer slot losses. On the other hand, the second frame structure should be used for static or quasi-static nodes. The proposed PHY allows switching between the two frame structures either through command line at initialization or during run-time in order to adapt its operation to different radio link conditions.

The Payload Data Unit (PDU) adopted by the proposed PHY is a Transport Block (TB), which is a concept reused from LTE PHY layer. Therefore, a TB is the payload coming from upper layers and given to PHY to be transmitted over the air through the PDSCH. A TB is defined in the LTE standard and varies according to some parameters [57]. According to the LTE standard, 1 TB consists of a number of bits that can be accommodated within a 1 ms long subframe given the selected number of RBs and MCS [57]. Therefore, given the number of allocated RBs and the desired MCS, upper layers can find the number of bits that can be handled by an 1 ms long slot.

The communication between upper layers, e.g., MAC, and the proposed PHY layer is carried out through the exchange of four messages. The first two, namely, Tx and Rx Control messages, are used to manage slot transmission and reception respectively. The parameters carried by these two messages can be configured and sent to PHY by upper layers before the transmission of every slot, hence allowing

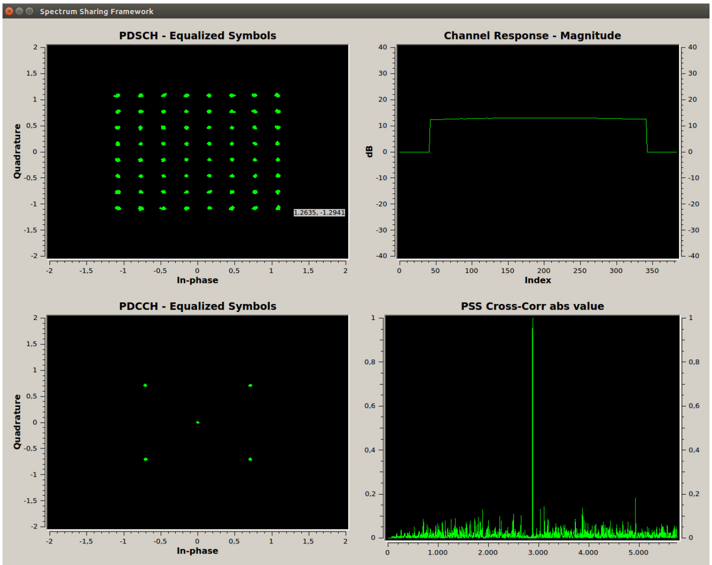


Figure 4.7: Slot-based PHY graphical user interface.

runtime configuration. The other two messages, namely, Tx and Rx statistics messages, are used to provide real-time feedback from PHY to upper layers, yielding vital information necessary for such layers to take actions.

Tx control messages carry the user data (i.e., TB) to be transmitted and Tx parameters related to that transmission, namely, number of RBs, MCS, data length, Tx gain, Tx channel, Tx PHY BW, transmission timestamp and frame structure type. The transmission timestamp parameter enables time-scheduled transmissions, which allows the implementation of Time Division Duplex (TDD) or TDMA technologies with the proposed slot-based PHY layer. **Rx control** messages are used to configure Rx channel, Rx gain and Rx PHY BW.

The other two messages, namely, Rx and Tx statistics, are used to inform upper layers of PHY Rx and Tx processing results respectively. **Rx statistics** messages carry the received data and reception statistics related to the received data such as CQI, RSSI, decoded MCS, slot error counter, decoding time, etc. **Tx statistics** messages inform upper layers of transmission statistics like coding time, total number of transmitted slots. Table 4.2 summarizes all the real-time configurable parameters and statistics offered by the slot-based PHY.

Figure 4.7 depicts the GUI provided by the slot-based PHY, which can be used for debugging or demonstration purposes. The GUI plots the constellation diagram of equalized and decoded PDSCH and PDCCH symbols (upper and lower left respectively), the equalized channel response (upper right) and the correlation peak of the PSS (lower right).

4.4.1 f-OFDM

As it is well-known, OFDM-based waveforms are not suited for spectral coexistence due to their poor spectral localization [58]. This problem is caused by the rectangular pulse-shape intrinsically employed in OFDM, which leads to a sync-pulse property in the frequency domain with a very low second lobe attenuation of -13 dB [59].

One of the simplest but still very efficient approaches used to guarantee better spectral localization, i.e., OOB emissions, and maintain the complex-domain orthogonality of the OFDM symbols is to apply some sort of filtering to the time domain OFDM symbols, giving rise to a new waveform known as f-OFDM [60]. Appropriate filtering of the OFDM signal must satisfy the following criteria: (i) have a flat pass-band over the subcarriers in the useful sub-band/subcarriers; (ii) have a sharp transition-band to minimize the necessary guard-bands; (iii) have sufficient (i.e., fair) stop-band attenuation; and (iv) a time duration that is comparable to a small fraction of an OFDM symbol's duration.

A FIR filter with a rectangular frequency response, i.e., a sinc-pulse impulse response, meets these criteria. To make this causal, the low-pass filter is realized using a window, which, effectively truncates the impulse response and offers smooth transitions to zero on both ends of it [61]. This filter causes no distortion in the pass-band and exhibits total out-of-band rejection. Therefore, the filter used in this work is defined in time domain as

$$f(n) = \frac{p(n)w(n)}{\sum_k p(k)w(k)}, \tag{4.1}$$

where $p(n)$ is the sinc impulse response and $w(n)$ is the truncation window (i.e., the window used to truncate the sinc signal). These two functions are respectively defined as

$$p(n) = \begin{cases} \sin\left(\frac{12\pi \text{NPRB} n}{\text{NFFT}}\right) / \frac{12\pi \text{NPRB} n}{\text{NFFT}}, & n \neq 0, \\ 1, & n = 0. \end{cases} \tag{4.2}$$

$$w(n) = \left\{ \frac{1}{2} \left[1 + \cos\left(\frac{2\pi n}{L-1} \right) \right] \right\}^{0.6}, \tag{4.3}$$

where NPRB is the number of physical resource blocks, NFFT is the FFT length used in the OFDM modulation, L is the filter length and $-\frac{(L-1)}{2} \leq n \leq \frac{(L-1)}{2}$.

The window defined in (4.3) exhibits smooth transitions to zero on both of it ends. These smooth shifts are very import to avoid abrupt transitions at the beginning and end of the truncated filter, $f(n)$, which in consequence, refrains the

[1] Depends on the USRP daughter board installed [26].

[2] The numbers correspond LTE BWs: 1.4, 3, 5, 10, 15 and 20 MHz respectively.

[3] The numbers correspond to frame structure types where PSS/SSS signals are added to all slots in a burst and PSS/SSS signals are only added to the first slot in a burst respectively.

truncated filter from spilling over adjacent frequency. A very good time/frequency localization is conferred to the impulse response of the truncated filter by applying the window, $w(n)$, to the Sinc function. Additionally, the windowing process keeps the ISIs resulting of the f-OFDM signal modulation within acceptable levels once the main filter's energy stays restricted to its main lobe in time domain [61].

As showed in Figure 4.8, the slot-based PHY creates its f-OFDM signal based on the modulation of M consecutive subcarriers in a number of consecutive OFDM symbols. The OFDM subcarriers are modulated with synchronization, reference and user/control data symbols (e.g., PSS/SSS sequences, QPSK/16QAM/64QAM and Reference symbols, etc.). During each OFDM symbol period, the PHY generates a N-point IFFT out of the M input symbols (i.e., the actual/useful OFDM symbol) and adds a CP to the front of the OFDM symbol. Note that N is the FFT size of the system, which is defined by the sampling rate and subcarrier frequency spacing, and that $N > M$. The unused OFDM subcarriers, $N - M$, are set to zero (i.e., *nulled*). The null subcarriers have two purposes: (i) the ones along both edges of the spectrum are used as guard-bands between adjacent channels, (ii) the one at DC, i.e., the central subcarrier, is used to allow the use of simple and cheap direct-conversion (zero intermediate frequency) receivers. Therefore, mathematically, the output signal of the slot-based PHY transmitter can be expressed as

$$s(n) = \sum_{l=0}^{L-1} s_l(n - l(N + N_g)), \tag{4.4}$$

with

$$s_l(n) \sum_{m=m'}^{m'+M-1} d_{l,m} e^{j2\pi mn/N}, \quad -N_g \leq n < N, \tag{4.5}$$

where N_g is the length of CP, $d_{l,m}$ is the symbol modulating m-th OFDM subcarrier of the l-th OFDM symbol, L denotes the number of OFDM symbols making up a subframe and $\{m', m' + 1, \cdots, m' + M - 1\}$ is the assigned subcarrier range. The f-OFDM signal is then obtained by passing the signal $s(n)$ through the designed spectrum shaping filter, $f(n)$, defined in (4.1), and defined as

$$\tilde{s}(n) = s(n) * f(n). \tag{4.6}$$

The spectrum shaping filter $f(n)$ has pass-band width equal to the total frequency width of the assigned subcarriers, i.e., the number of allocated RBs, and its time duration is only a fraction of the duration of an OFDM symbol. The filter duration is chosen in order to maintain OFDM's good time-localization feature [48].

Therefore, a FIR filter meeting the necessary requirements is added to the Tx processing flow, just after the OFDM modulation. The filter is applied just before the OFDM symbols are sent to the hardware, e.g., USRP. An efficient FIR is implemented on software making use of SIMD instructions present on most of the general purpose microprocessors, i.e., CPUs, available today to exploit the data-level parallelism present in the filtering processing. In Appendix A.1, we present

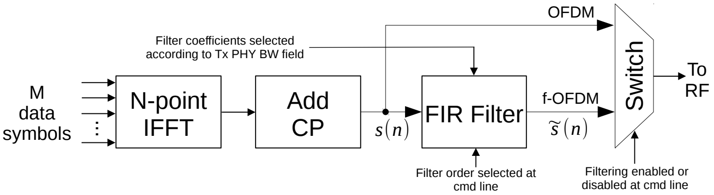

OFDM
Filter coefficients selected according to Tx PHY BW field
M data symbols
N-point IFFT
Add CP
$s(n)$
FIR Filter
f-OFDM
$\tilde{s}(n)$
Switch
To RF
Filter order selected at cmd line
Filtering enabled or disabled at cmd line

Figure 4.8: f-OFDM enabled transmitter.

an example of an optimized digital FIR filter implementation using Advanced Vector Extension (AVX) instructions. Our comparisons with a naive implementation show that such SIMD instructions substantially speed-up the filtering processing. The filter's coefficients applied to the modulated OFDM signal are automatically selected according to the configured Tx PHY BW (i.e., the coefficients are selected in real-time based on the Tx PHY BW field in the Tx control message) as it needs to have its cut-off frequency changed to exactly filter the desired signal's BW. Figure 4.8 shows the f-OFDM enabled transmitter scheme, which can have the FIR filtering processing enabled or disabled and the filter order selected at command line during start up of the framework.

It is interesting pointing out that f-OFDM is MIMO-friendly in the sense that it can be easily integrated with (massive) MIMO implementations, not requirement complex modifications. MIMO or massive MIMO can help a system achieve better spatial localization, which means the narrow beams created by the large number of antennas do not overlap between them, decreasing interference between the beams. In [49], we showed that the interference caused by users with f-OFDM PHYs transmitting at closely separated adjacent channels can be mitigated by having a BS equipped with a large number of antennas. As the number of antennas increases, the interference and noise tend to vanish as the devices' channels become asymptotically orthogonal due to the law of large numbers.

4.4.2 Advantages of the proposed slot-based PHY

In this section we describe the main advantages offered by the proposed slot-based PHY. The first advantage is that there is no need to align the start of transmissions to subframe boundaries. This is in contrast to LTE-LAA, where reservation signals are transmitted in order to prevent other RATs (e.g., Wi-Fi) from initiating a transmission until the next subframe boundary. The transmission of reservation signals clearly reduces the utilization efficiency of the available radio resources. As the proposed PHY adopts bursty transmissions, all the information required for synchronization and demodulation is self-contained in every slot. This allows for the development of standalone solutions that do not need to rely on reservation signals nor be subframe-aligned to an anchor on licensed spectrum.

A second advantage of the proposed PHY is that the filtering applied to the OFDM symbols makes the f-OFDM PHY more spectral efficient as the OOB emis-

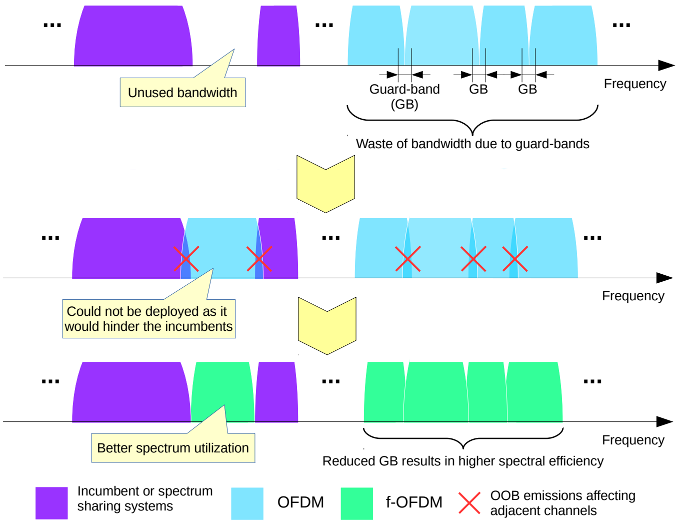

Unused bandwidth
Guard-band (GB)
GB
GB
Frequency
Waste of bandwidth due to guard-bands
Could not be deployed as it would hinder the incumbents
Frequency
Better spectrum utilization
Frequency
Reduced GB results in higher spectral efficiency
Incumbent or spectrum sharing systems
OFDM
f-OFDM
OOB emissions affecting adjacent channels

Figure 4.9: Closer coexistence to other RATs due to reduced OOB emissions.

sions are reduced. To be more specific, in LTE, 10% of the allocated BW is reserved as guard band, which allows the waveform signal to attenuate and therefore, meet the spectrum mask requirements [62]. Undoubtedly, this is a considerable waste of frequency resources, which are becoming more and more precious. Additionally, OOB emissions interfere with other closed (i.e., nearby) allocated systems, decreasing the quality of the received signal, which impacts on the throughput experienced by that system. The reduced OOB emissions make the proposed PHY ideal for coexistence with other RATs (e.g., incumbent systems, spectrum sharing RATs, etc.), allowing it to operate closer to other systems in frequency domain and consequently, reducing spectrum wastage while increasing the spectral efficiency as shown in Figure 4.9.

Another advantage, offered by the proposed framework, is the possibility to configure in real-time all PHY (and also RF Monitor and LBT modules) parameters through pre-defined control messages, which can also be easily modified (thanks to the use of Google's protobuf interface description language) to accommodate new parameters. By plugging in a new module to the system, it is possible to experiment with the PHY parameters in a programmatic way, so that new algorithms can be developed and tested.

4.5 RF Monitor

As mentioned earlier, this module runs on an exclusive thread and utilizes Radio # 1 of the USRP to act as a spectrum sensing monitor. It monitors and reports on a defined range of spectrum band and can be configured to report either the detected

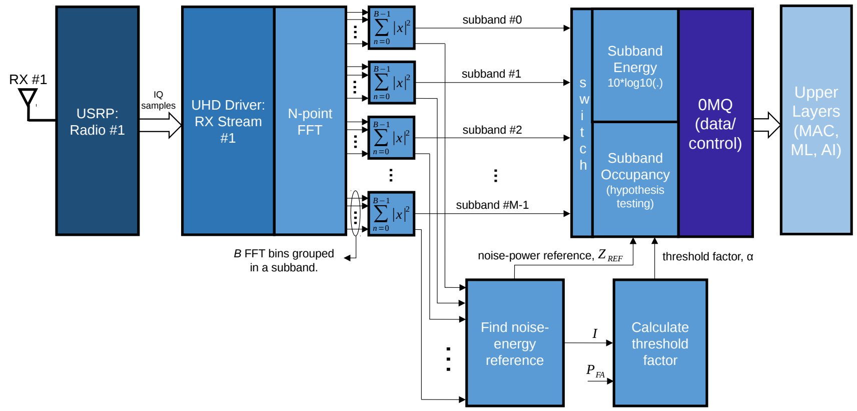

RX #1
USRP: Radio #1
IQ samples
UHD Driver: RX Stream #1
N-point FFT
$\sum_{n=0}^{B-1} |x|^2$
subband #0
subband #1
subband #2
subband #M-1
s w i t c h
Subband Energy 10*log10(.)
Subband Occupancy (hypothesis testing)
0MQ (data/ control)
Upper Layers (MAC, ML, AI)
B FFT bins grouped in a subband.
noise-power reference, Z_{REF}
threshold factor, α
Find noise-energy reference
I
P_{FA}
Calculate threshold factor

Figure 4.10: RF Monitor architecture.

power per subband or subband occupancy (statistical hypothesis testing [63]), i.e., if a subband is free or already being occupied by another transmitter. Figure 4.10 depicts the architecture of the RF Monitor module.

Subband power reporting is obtained by applying a N-point FFT to the received IQ samples, summing the squared modulus of every consecutive B FFT bins (where B must be a power of two) to form M groups of power samples, i.e., subbands and then, applying the base-10 logarithm in relation to a reference power of 1 Watt to each one of the subbands, resulting in measurements given in dBW. A M-long float array holds the subband power information.

In order to report subband occupancy, the module implements a spectrum-sensing algorithm based on a CA-CFAR strategy [63, 64]. The proposed algorithm is composed of three stages: (i) identification of a set of power samples, I, that can be considered as containing only the presence of noise in order to calculate a noise-power reference value, Z_{REF}; (ii) the number of power sample values considered for the calculation of the noise-power reference value, and the desired probability of false alarm, P_{FA}, are used in the second stage to calculate the detection factor, α; (iii) the third stage employs both, the threshold factor and the noise power reference value to test the occupancy of each one of the subbands.

The power samples used in the first stage are obtained after applying an N-point FFT to the received IQ samples and summing the squared modulus of every B FFT bins into subbands. The procedure used to find the noise-power reference value, Z_{REF}, in the first stage involves ordering the subbands in ascending order of power. The detection threshold is found by multiplying Z_{REF} by the threshold factor, α. In the end, there are M subbands which will be tested against the detection threshold, in order to declare whether the subband is occupied or free. If the subband power is greater than the detection threshold, then the subband is declared occupied (true), otherwise it is declared free (false). After testing all subbands, an M-long boolean array holds the occupancy information of the monitored spec-

Table 4.3: RF Monitor real-time configurable parameters and statistics.

Message	Parameter	Type	Unit	Range
RF Monitor control	Monitored BW	float	Hz	depends on HW [1]
	Center frequency	float	Hz	depends on HW [1]
	Rx gain	uint32	dB	depends on HW [1]
	Number of FFT bins	uint32	-	16-16384
	Report periodicity	uint32	μs	≥ 150
RF Monitor statistics	IQ samples timestamp	uint64	s	≥ 0
	Power/Occupancy array	float[]/bool[]	-	float range or True/False

Table 4.4: Round Trip Time between UHD and x310 USRP device [70, 71].

Link Type	round trip time (RTT) [us]	Host PC configuration
PCIe	79	Intel i7-6700 3.4GHz, NI PCIe x4 card
10 Gbps ETH	106	Intel E5-2650 v4 2.2GH, Qlogic 57810 ETH
1 Gbps ETH	101	Intel i7-6700 3.4GHz, Intel i219-v ETH

trum band. In the current implementation, the probability of false alarm, P_{FA} is set to 10^{-4}. Further information on the subband occupancy algorithm is provided in Appendix B.1.

The module allows configuration of the following parameters for both types of reporting (power or occupancy): monitored BW (through the change of the sampling rate), center frequency, number of FFT bins M, which changes the frequency resolution, Rx gain, the number of consecutive FFT bins, B, considered as a subband (i.e., number of bins considered for the subband power calculation) and the periodicity of the statistics report. These parameters can be configured in real-time by upper layers through the **RF Monitor control** message. The sensing reporting type, i.e., power or occupancy, is configured during initialization of the framework. The following information is sent to upper layers through an **RF Monitor statistics** message: timestamp of the moment the IQ samples were received by the USRP and power or occupancy array depending on the configured reporting type. Table 4.3 summarizes all the real-time configurable parameters and statistics offered by the RF monitor module.

The RF monitor module is of great importance to spectrum sharing mechanisms as it offers a local insight of the spectrum band usage. It allows upper layers to access spectrum sensing measurements, which can be used to train ML and AI modules to better understand the environment, optimize the spectrum usage/sharing and cooperatively work with other networks without any previous knowledge on the other network's operation and implementation, i.e., without any co-design [65]. For instance, this module allows the implementation of adaptive carrier selection algorithms such as CSAT [66] and can also be used to train ML or AI modules to perform MF-TDMA slot allocation [38].

The RF Monitor module reports sub-band occupancy based on either hypothesis testing or simple energy calculation it performs on every sub-band. For instance, it can be used to monitor the whole available spectrum and transmit selectively on different sub-bands, i.e., channels, which are the same length as the

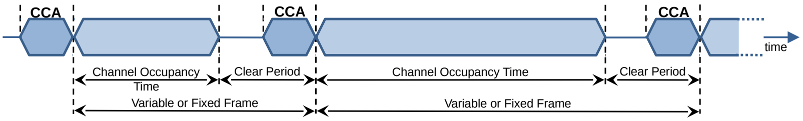


Figure 4.11: SW-based LBT frame structure.

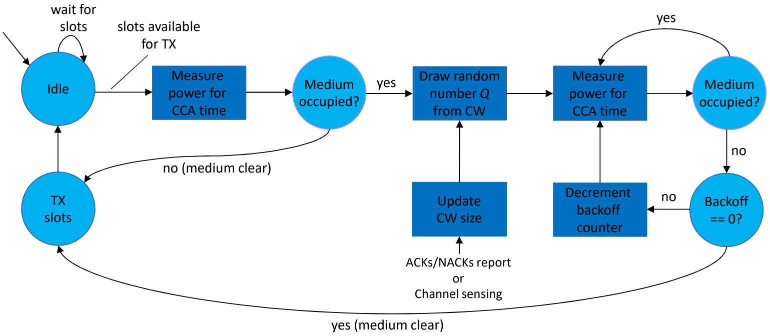


Figure 4.12: Software-based LBT Finite State Machine (FSM).

PHY's BW. For that, PHY BW can be reconfigured depending on the available spectrum, and the monitored sub-bands (i.e., channels) could have the same width as the configured PHY BW. Therefore, based on sub-band occupancy information, the system (i.e., MAC/AI layers) could select the best channel for a given transmission.

4.6 Listen Before Talk

As the other modules in the proposed framework, the software-based Listen Before Talk (SW-based LBT) module also runs on an exclusive thread. The module implements a contention-based mechanism for medium access based on 3GPP's Licensed Assisted Access (LAA) specifications [67, 68]. However, differently from 3GPP's LAA specifications, the proposed implementation allows the following configurations: minimum and maximum CW sizes (with the option for the upper limit to be fixed or dynamically adjusted), variable frame period (with configurable maximum COT and clear periods, which allows longer than the standardized maximum frame period [67] and the adaptation to different traffic loads and channel usage [69]) and threshold. Figure 4.11 shows the proposed variable frame structure supported by the implemented LBT module. The frame period, which is composed of the COT and clear periods, can be dynamically configured to any value greater than 1 ms.

The rationale behind the decision to implement a SW-based LBT module is explained next. We start by presenting in Table 4.4 RTT measurements for an x310 USRP device combined with different communication links. Notice that, each la-

Table 4.5: LBT real-time configurable parameters and statistics.

Message	Parameter	Type	Unit	Range
LBT control	Sampling rate	float	Sps	depends on HW [1]
	Center frequency	float	Hz	depends on HW [1]
	CCA threshold	float	dBW	float range
	CCA observation time	uint32	μs	> 0
	CW type	uint8	-	0/1 (fixed/dynamic)
	Minimum CW value	uint32	-	≥ 0
	Maximum CW value	uint32	-	≥ 0
	Idle time	uint32	μs	≥ 0
LBT statistics	Channel clear ratio	float	-	0-100 %
	Channel clear avg. energy	float	dBW	float range
	Channel occupied ratio	float	-	0-100 %
	Channel occupied avg. energy	float	dBW	float range

tency measurement is calculated as the average over 10 trials for each one of the following sampling rates: 5, 10 and 25 MHz. The measurements show that compared to the 1 ms long slot of the proposed PHY, it is feasible to use a SW-based LBT implementation as the RTT is of approximately 100 [us]. The SW-based implementation avoids the development of an FPGA-based LBT module, which allows the framework to be used and ported to a wider range of SDR devices. Additionally, the SW-based implementation makes the LBT module easier to be customized by a wider range of users, as FPGA development is not so widespread and straightforward as software development.

Figure 4.12 depicts the FSM of the proposed SW-based LBT module, which is explained as follows. Initially, a radio having data to transmit performs a Clear Channel Assessment (CCA) in order to determine if the medium is occupied or clear, i.e., the radio performs energy detection (ED) in given spectrum band during CCA observation time, if the measured energy level exceeds the CCA threshold then the medium is declared as occupied, otherwise, the medium is clear. If the radio finds the medium clear, it may transmit immediately and occupy the medium for the total time of COT.

In case the medium is occupied, the radio must execute a random backoff procedure, where a backoff counter is set to a random number, Q, drawn from a specified interval called the CW [72]. During the random backoff procedure, Q CCA checks are performed for the duration of a CCA observation time, where Q defines the total number of clear periods that need to be observed (i.e., counted) before the radio can transmit. The backoff counter is decremented every time the medium is observed to be clear. The radio may transmit when the counter reaches zero. The random backoff procedure was designed to cope with situations where more than one radio senses the medium as being in the clear state (i.e., there is no transmission occurring at that time) at the same time, and with this, decreasing the collision probability.

The CW size can be configured to be fixed or dynamic, with adjustable minimum and maximum values. In the fixed mode, Q is always drawn from the same

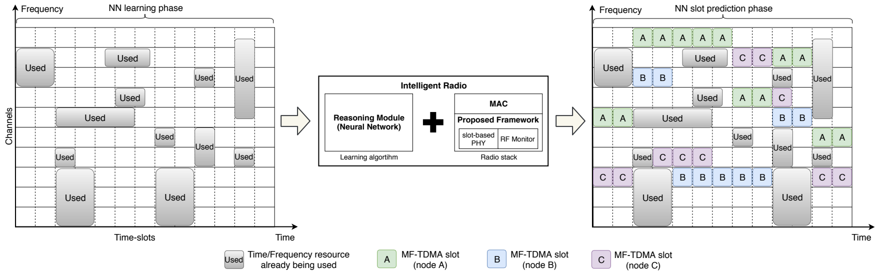


Figure 4.13: Use case scenario: MF-TDMA free slot prediction.

interval, giving different radios the same probability of transmission opportunity in spite of collisions and traffic load. This approach is not fair when coexisting with RATs that adopt exponentially increasing CWs (e.g., Wi-Fi) when a collision happens. In the dynamic mode, a binary exponential backoff mechanism is used in the way that the CW size is increased exponentially based upon the occurrence of a collision (which can be based on feedback reports such as HARQ Acknowledgment (ACK)/NACK or based on channel utilization sensing [73]) and reset to the minimum value when the transmission succeeds. The dynamic mode was devised to improve the overall channel utilization, reduce the collision probability and fairly coexist with other RATs like Wi-Fi.

The following parameters can be configured in real-time by upper layers through an **LBT control** message: sampling rate, center frequency, CCA threshold, CCA observation time, CW type (fixed or dynamic), CW minimum and maximum values and idle time. Additionally, the proposed LBT design keeps track of the following channels' occupancy statistics: channel clear/occupied ratios and clear/occupied average energies. These statistics are reported to upper layers through an **LBT statistics** message and can be used, for example, to find the optimum channel to be used, i.e., the least occupied channel. Table 4.5 summarizes all the real-time configurable parameters and statistics offered by the LBT module.

4.7 Use Cases

In this section we describe three use cases for the proposed framework. The first one focuses on how the framework may facilitate slot selection in time and frequency domains in order to coexist with other technologies, based on intelligent prediction of the medium usage. The second use case shows how the framework can allow contention-based parameters at MAC level to be optimized when an intelligent upper layer is integrated for specific technologies. The third one shows how the framework can be used to devise and test flexible spectrum sharing paradigms.

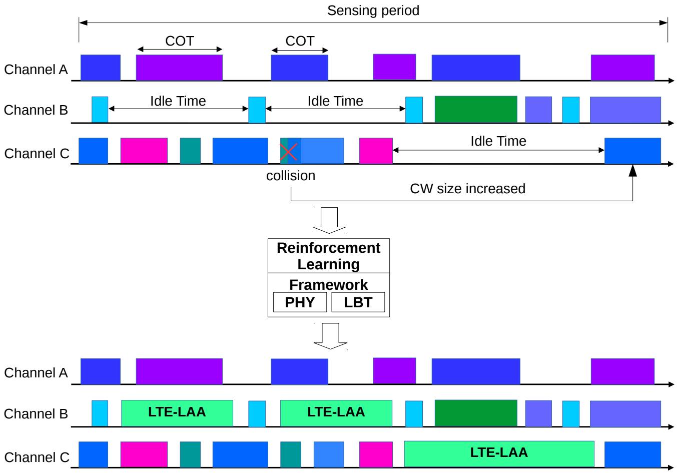


Figure 4.14: Use case scenario: Intelligent RAT Coexistence.

4.7.1 Transmission Pattern Prediction

In the unlicensed 5 GHz spectrum band, RATs such as LTE-U, LTE-LAA and Wi-Fi use the same spectrum for communications, which often results in cross-technology interference, i.e., interference from spatially close concurrent transmissions that overlap in time and frequency, an event known as collision.

In order to mitigate cross-technology interference, in this use case, the proposed framework is used to implement an intelligent MF-TDMA like network in a congested unlicensed spectrum. Here, the slot-based PHY and the RF Monitor modules are integrated with a MAC layer and a reasoning module featuring an NN model.

The NN is trained to improve slot scheduling (i.e., time/frequency allocations) by predicting transmission patterns, and consequently avoiding traffic from other nodes regardless of the technologies they are using. By observing the spectrum, through statistics received from both the RF Monitor and PHY modules, the NN is able to do online learning and predict the behavior of other nodes, finding free slots some time in advance. Based on the slot usage predictions provided by the reasoning module, the MAC layer can select in advance the best channel and the necessary COT and MCS to deliver the traffic load while not causing interference (i.e., collisions) to other radios. Figure 4.13 shows how the integration of the proposed framework with a reasoning module can be used to implement an intelligent spectrum sharing scheme.

It is important to notice that any type of NN model (e.g., Deep Convolutional Networks, auto-encoders, Support Vector machines, etc.) could be employed in this use case, as the framework is totally independent of the approaches (i.e., software and/or hardware) used in the upper layer's implementation.

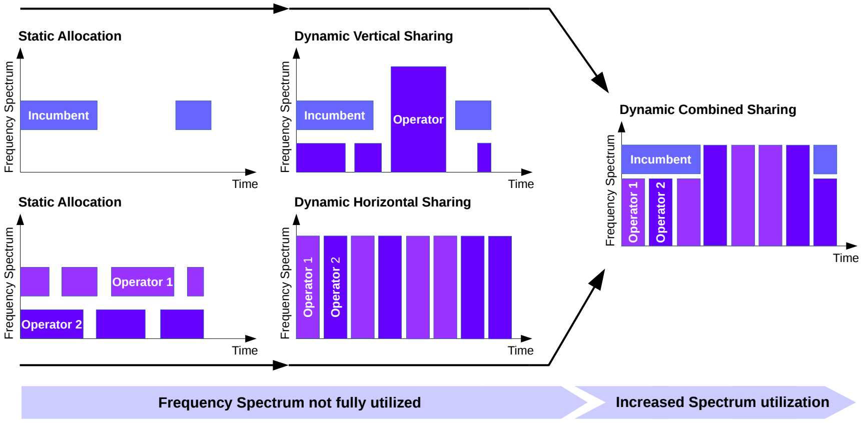


Figure 4.15: Use case scenario: Coordinated Spectrum Sharing Schemes.

4.7.2 Intelligent LTE-LAA and Wi-Fi Coexistence

LTE-LAA enables the deployment of LTE networks in the unlicensed 5 GHz spectrum band. However, when coexisting with LTE-LAA, the performance of Wi-Fi networks primarily relies on how the LTE-LAA parameters are configured, especially parameters such as channel, COT, idle time and CW size that govern the transmission opportunities in LTE-LAA networks.

In order to provide fairer coexistence (i.e., provide fair opportunities for all technologies to access the medium), LTE-LAA parameters must be dynamically configured based on specific statistics like collision counters, different channel utilizations, i.e., the traffic load present in a channel, etc. In this use case, the proposed framework is used to implement an intelligent LTE-LAA BS. Here, the slot-based PHY and the LBT modules are integrated with a Reinforcement Learning (RL) module, which can feature algorithms like Multi-Armed Bandit or Q-Learning, to maximize the overall capacity performance through an efficient coexistence.

The RL module learns how to estimate the activity of Wi-Fi users and consequently adjust the LTE-LAA parameters for several traffic conditions. The main goal is to determine a policy by which the LTE-LAA BS can intelligently choose the optimum channel, COT, idle time and CW size based on collision statistics and measurements taken during the sensing period. Figure 4.14 shows how the integration of the proposed framework with an RL module can be used achieve efficient RAT coexistence. In the figure, we see in the upper part the sensing period, where the RL module learns about the medium usage, i.e., how often and how long others networks access the channels. Based on what the module learned, we see in the lower part of the figure that the LTE-LAA node is able to select channels that can accommodate its traffic load needs.

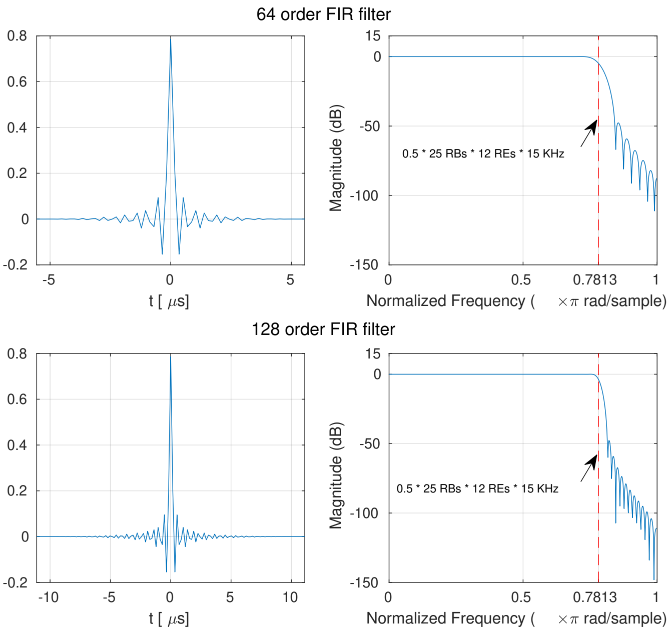


Figure 4.16: Impulse and frequency responses of the designed filter for f-OFDM with two different filter orders and BW equal to 25 RBs.

4.7.3 Coordinated Spectrum Sharing Schemes

The adoption of spectrum sharing mechanisms is expected to bring critical benefits to the next generation of wireless networks. For example, spectrum sharing can unlock additional, currently underutilized, spectrum BWs, make it possible to introduce new deployment scenarios and improve spectrum utilization [74, 75].

There exists two main approaches for spectrum sharing, namely, uncoordinated and coordinated. Uncoordinated approaches make use of contention-based mechanisms like the LBT in LTE-LAA and Carrier-sense multiple access with collision avoidance (CSMA/CA) in Wi-Fi. However, at high traffic loads, as users (i.e., radios) are always competing for access to the medium without any cooperative access planning and organization, the number of collisions can be very high, which increases the radios' backoff period and as consequence, drastically reduces their throughput. Therefore, such approach is far from optimal in terms of spectrum efficiency and QoS guarantees. The low spectrum efficiency and the lack of QoS guarantees are its main disadvantages.

On the other hand, coordinated spectrum sharing makes use of a centralized database or over-the-air protocols to define a cooperative access planning and organization among users, i.e., radios. The efficiency and robustness of spectrum sharing users can be hugely increased by exploring tight coordination and time synchronization among them. Coordinated sharing has the opportunity to intro-

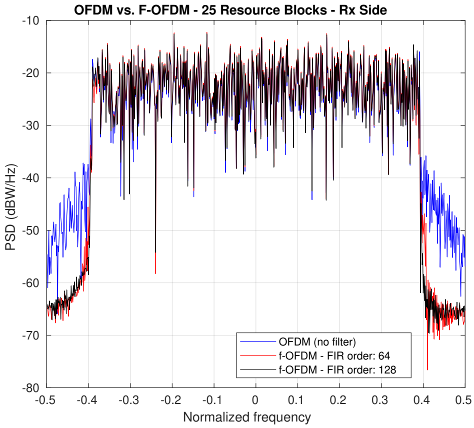


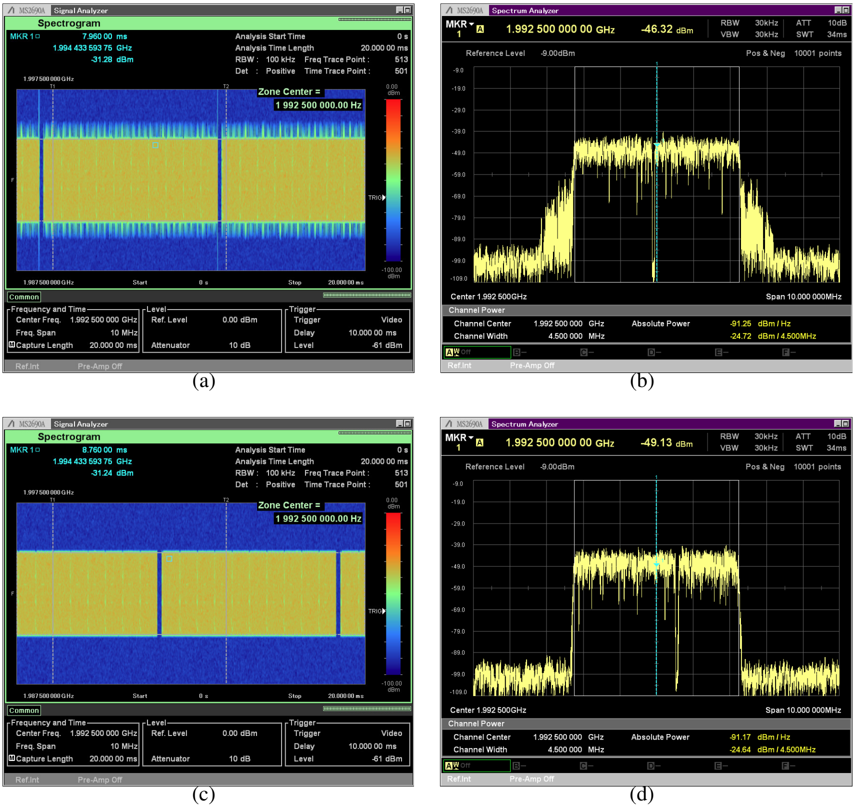
Figure 4.17: OFDM and f-OFDM OOB emissions comparison at Rx side.

duce new sharing paradigms such as vertical, horizontal and dynamic (combined) spectrum sharing, as depicted in Figure 4.15. By analyzing Figure 4.15, it is clear that a better spectrum utilization can be achieved by a dynamic spectrum sharing coordination among users. The tight coordination of time and frequency resources among users can deliver QoS levels similar to that of users with exclusive allocations of spectrum BW. Through coordinated spectrum sharing, primary users (e.g., incumbents) can get access to guaranteed resources (i.e., access to the medium), while any spectrum not being used by them can be dynamically allocated among secondary users.

In this use case, the slot-based PHY with its real-time configurable parameters (e.g., Tx PHY BW, transmission timestamp, COT) can be used to devise and assess the performance of new and disruptive sharing protocols and models, deployment scenarios, etc. For example, Game theoretic models can be used to solve spectrum management problems like spectrum trading, spectrum sharing, interference avoidance, power allocation, etc. [76, 77]. In this use case, a central unit, employing a Game theoretic model, would receive requests for time and frequency resource allocation from highly heterogeneous users with different QoS demands and based on some QoS criteria like average throughput, delay, etc. the central unit would try to maximize the system-wide QoS following some pricing mechanism for resource allocation [77], i.e., the central unit optimizes the spectrum resources in order to meet the users' QoS requirements [78].

4.8 Experiment Results

In this section we present some experimental results in order to demonstrate the effectiveness and usability of the proposed framework. All the experiments presented here were carried out with the framework running on servers with Intel



(a)

MKR 1 7.960 00 ms
1.994 433 593 75 GHz
-31.28 dBm
Analysis Start Time 0 s
Analysis Time Length 20.000 00 ms
RBW : 100 kHz Freq Trace Point : 513
Det : Positive Time Trace Point : 501
Zone Center = 1 992 500 000.00 Hz
Start 0 s Stop 20.000 00 ms
Frequency and Time: Center Freq. 1.992 500 000 GHz; Freq. Span 10 MHz; Capture Length 20.000 00 ms
Level: Ref. Level 0.00 dBm; Attenuator 10 dB
Trigger: Trigger Video; Delay 10.000 00 ms; Level -61 dBm
Ref.Int Pre-Amp Off

(b)

MKR 1 1.992 500 000 00 GHz -46.32 dBm
RBW 30kHz VBW 30kHz ATT 10dB SWT 34ms
Reference Level -9.00dBm
Pos & Neg 10001 points
Center 1.992 500GHz Span 10.000 000MHz
Channel Power
Channel Center 1.992 500 000 GHz
Channel Width 4.500 000 MHz
Absolute Power -91.25 dBm / Hz
-24.72 dBm / 4.500MHz
Ref.Int Pre-Amp Off

(c)

MKR 1 8.760 00 ms
1.994 433 593 75 GHz
-31.24 dBm
Analysis Start Time 0 s
Analysis Time Length 20.000 00 ms
RBW : 100 kHz Freq Trace Point : 513
Det : Positive Time Trace Point : 501
Zone Center = 1 992 500 000.00 Hz
Start 0 s Stop 20.000 00 ms
Frequency and Time: Center Freq. 1.992 500 000 GHz; Freq. Span 10 MHz; Capture Length 20.000 00 ms
Level: Ref. Level 0.00 dBm; Attenuator 10 dB
Trigger: Trigger Video; Delay 10.000 00 ms; Level -61 dBm
Ref.Int Pre-Amp Off

(d)

MKR 1 1.992 500 000 00 GHz -49.13 dBm
RBW 30kHz VBW 30kHz ATT 10dB SWT 34ms
Reference Level -9.00dBm
Pos & Neg 10001 points
Center 1.992 500GHz Span 10.000 000MHz
Channel Power
Channel Center 1.992 500 000 GHz
Channel Width 4.500 000 MHz
Absolute Power -91.17 dBm / Hz
-24.64 dBm / 4.500MHz
Ref.Int Pre-Amp Off

Figure 4.18: Comparison between OFDM and f-OFDM. (a) Spectrogram with non-filtered OFDM. (b) Spectrum with no filter enabled. (c) Spectrogram with 128 order FIR filter enabled. (d) Spectrum with 128 order FIR filter enabled.

Xeon E5-2650 v4 CPUs (@2.2 GHz,30 M cache, 9.60 GT/s QPI, Turbo, HT, 12 Cores/24 Threads, 105 Watts) with 128 Giga Bits of RAM memory connected to x310 USRPs with 10 Gigabit Ethernet links, and equipped with CBX-120 RF daughterboards [79]. These RF daughterboards operate from 1200 up to 6000 MHz with a BW of 120 MHz [79]. A PHY BW of 5 MHz is selected for the experiments presented in this section, which in frequency corresponds to 25 RBs * 12 subcarriers * 15 KHz (subcarrier separation in frequency domain) = 4.5 MHz of useful BW, i.e., the BW really occupied by the transmission of data.

4.8.1 Experiment with f-OFDM waveform

Next we present a few comparison results showing how OOB emissions can be mitigated with the use of a FIR filter added to the PHY Tx processing chain, just

Table 4.6: OFDM vs. f-OFDM OOB emission and processing time comparison.

FIR order	**PSD @ 0.4*fs* [dBW/Hz]**	**PSD @ 0.5*fs* [dBW/Hz]**	**Processing time [ms]**
No Filter	-40.83	-56.21	-
64	-52.12	-64.52	0.12
128	-55.39	-65	0.24

after the OFDM symbol modulation. The spectrum shaping filters employed in the following experiments are obtained by windowing the Sinc function, $p(n)$, as defined in (4.2) with the window function, $w(n)$, given in (4.3).

Figure 4.16 shows the base-band impulse response of the designed filter with BW (i.e., NPRB) equal to 25 RBs and NFFT equal to 384. The upper and lower parts of the figure show the impulse and frequency responses of 64-th and 128-th order filters respectively. It can be noticed that the main energy of the filters is confined within the the main lobe, centered around 0 μs, and consequently, the resulting ISIs stays within tolerable levels. The figure also shows the 3 dB cutoff frequency (red-dashed lines) of the filters, which, as designed, happens around half of the useful BW, i.e., 4.5 MHz / 2 = 2.25 MHz. As expected, the 128-th order filter presents a steeper transition region, which results in less interference to adjacent channels and a better frequency-localization when compared with OFDM.

First, we present some results collected at the Rx side of the proposed slot-based PHY. Figure 4.17 depicts the PSD of the f-OFDM signal and compares it with an OFDM signal. The figure shows the comparison of OOB emissions between the OFDM and f-OFDM waveforms at the receiver side of the slot-based PHY for two different FIR filter orders, 64 and 128 respectively. For this result, the FIR filter, either with order 64 or 128, is added to the Tx processing flow (i.e., OFDM symbol modulation), see Figure 4.8. The figure is obtained by collecting IQ samples at the receiver side of the slot-based PHY, after Rx front end-processing (RF and digital processing) and after the slot (i.e., subframe) is synchronized and aligned, therefore, this is the signal fed into the demodulation module at the receiver side of the slot-based PHY. Table 4.6 compares some approximate measurements of PSD and filter processing time.

As can be seen by analyzing Figure 4.17 and Table 4.6, the filter reduces the OOB emissions from around -40 dBW/Hz to less than -52 dBW/Hz at the edge of the transmitted signal, around 0.4*fs*, where *fs* is the sampling rate. As can also be noticed in Figure 4.17, the 128 order FIR filter has, as expected, a sharper transition region when compared to the 64 order FIR filter. Table 4.6 also shows the processing time for each one of the FIR filter orders. The processing time presented in the table considers the time it takes to filter a 1 ms long slot. The values are the result of the average of 10000 slot transmissions. For a PHY with 25 RBs of BW, a 1 ms long slot is composed of 5760 complex samples, i.e., 5760 I/Q samples. Each complex sample is represented by 32 bits (i.e., 4 bytes, with 2 bytes representing each one of the I and Q components), which means the filter

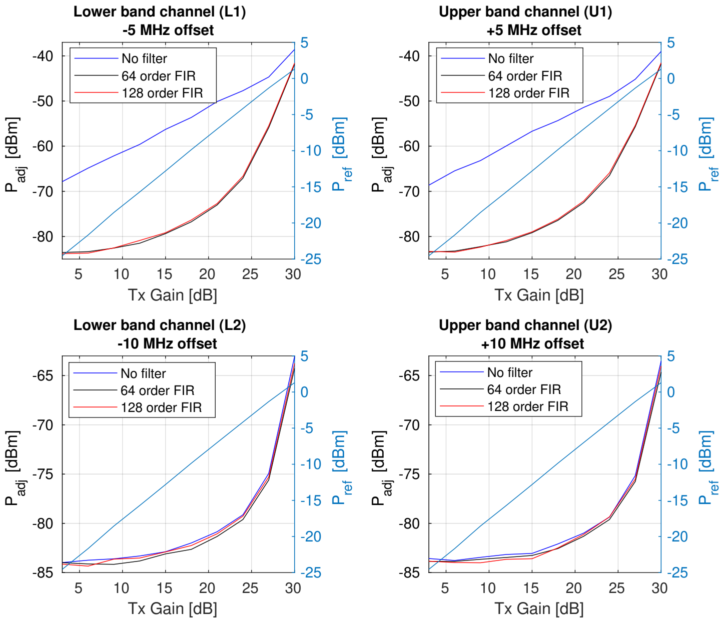


Figure 4.19: Adjacent Channel Leakage Ratio (ACLR) measurements for different Tx gains and filter orders.

has to process 23040 bytes.

In Figure 4.18 we compare the spectrogram and spectrum of OFDM and f-OFDM transmissions collected with an Anritsu MS2690A Signal Analyzer. The figures were collected with a Tx center frequency of 1.9925 GHz, Tx gain of 3 dB with the USRP Tx output connected to the signal analyzer through a cable with 20 dB of attenuation. It is easily seen that the OFDM side lobes (i.e., OOB emissions) are reduced with the use of the 128 order FIR filter.

The USRP expects to be fed with samples at a rate close to *fs*, however, if the processing time of the Tx chain increases so that it can not deliver samples fast enough (i.e., at a rate close to *fs*), then the USRP starts outputting 'U's (Underflows) at the command line, meaning, that the slot-based PHY transmitter is not supplying samples fast enough. Therefore, in order to keep the Tx processing flow close to the specified sample rate, the processing time of the filter must be as low as possible. During the experiments with the current setup, we also tested higher order filters like 256 and 512, however, the processing time increased to the point where the slot-based PHY transmitter was not being able to supply samples fast enough to the USRP. For smaller PHY BWs the number of complex samples making up a slot is smaller, which makes it possible to use higher order filters. Other FIR filter orders can be easily added to the slot-based PHY by just updating the table with filter coefficients.

Another kind of measurement that was taken in order to show the improvement

Table 4.7: Difference between P_{adj} (no filter) and P_{adj} (64/128 order filters).

Offset	**L1 (-5 MHz)**		**U1 (+5 MHz)**		**L2 (-10 MHz)**		**U2 (+10 MHz)**		**Tx gain [dB]**
FIR order	**64**	**128**	**64**	**128**	**64**	**128**	**64**	**128**	
Difference [dB]	15.69	15.97	14.82	14.66	0.01	0.16	0.29	0.27	3
	18.5	18.82	17.77	18	0.38	0.6	0.09	0.21	6
	20.43	20.39	19.06	19.17	0.56	0.03	0.20	0.55	9
	21.88	21.27	21.24	20.95	0.51	0.21	0.30	0.49	12
	23.09	22.89	22.47	22.28	0.22	0.02	0.21	0.54	15
	23.11	22.71	22.1	21.84	0.65	0.26	0.47	0.40	18
	22.9	22.59	21.07	20.73	0.48	0.21	0.32	0.13	21
	19.38	18.94	17.52	16.94	0.48	0.14	0.25	-0.02	24
	11.21	10.8	10.51	10.19	0.67	0.36	0.6	0.35	27
	3.29	3.02	2.74	2.51	1.18	0.68	1.12	0.51	30

provided by the f-OFDM waveform is known as ACLR. ACLR is an important characteristic of wireless transmitters and is defined as an important wireless metric by national laws regulating radio standards as well as by 3$^{\text{rd}}$ Generation Partnership Project (3GPP) and Institute of Electrical and Electronics Engineers (IEEE).

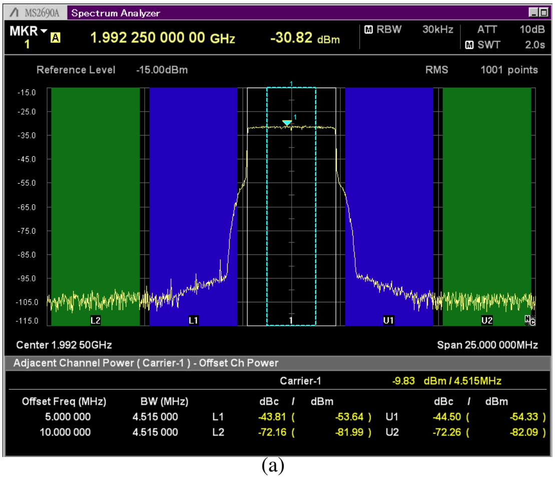
ACLR is defined as the ratio between the total integrated adjacent channel average power, P_{adj}, centered at upper and lower channel frequencies and the total integrated reference channel average power, P_{ref}, centered at the assigned reference channel frequency (i.e., in-band power), where the powers are measured after a receiver filter [80]. The ACLR metric is defined as

$$\text{ACLR(dBc)} = 10\log_{10}\left(\frac{P_{\text{adj}}}{P_{\text{ref}}}\right) = P_{\text{adj}}(\text{dBm}) - P_{\text{ref}}(\text{dBm}). \tag{4.7}$$

In digital communications systems, the power that leaks from a transmitted signal into adjacent channels can interfere with transmissions in the neighboring channels and decrease the neighbor system's performance. For LTE, the ACLR metric verifies that transmitters are performing within specified limits defined in the 3GPP specifications [81, 82].

Figure 4.19 presents the adjacent channel average power, P_{adj} in dBm, for different frequency offsets, Tx gain values and FIR filter orders. The figure also shows the reference channel average power P_{ref} in dBm. The measurements were also taken with the Anritsu MS2690A Signal Analyzer for a setup where Tx center frequency is set to 1.9925 GHz, COT is set to 100 ms, which means 100 1 ms-long slots are transmitted in a row, with an idle time of 1 ms between consecutive transmissions and MCS is set to 0. The Tx gain is changed from 3 up to 30 dB in steps of 3 dB for three different cases: no filter, 64 and 128 order FIR filters. The connection between the USRP and the Spectrum Analyzer is done through cable and attenuation of 20 dB is applied to the signal.

Table 4.7 shows the difference between P_{adj} when no filter is used and P_{adj} when 64 or 128 order FIR filters are used. As can be seen, the power leakage on adjacent channels with frequency offset of +/- 5 MHz is greatly decreased when using any of the two filter orders. For these cases, the leakage reduction can be as



MKR 1 1.992 250 000 00 GHz -30.82 dBm RBW 30kHz ATT 10dB SWT 2.0s

Reference Level -15.00dBm RMS 1001 points

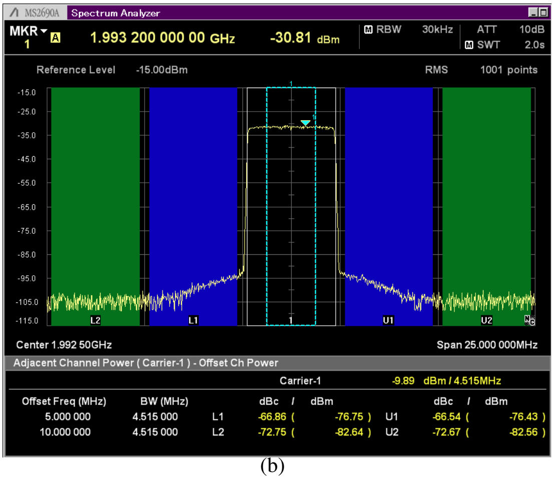
Center 1.992 50GHz Span 25.000 000MHz

Adjacent Channel Power (Carrier-1) - Offset Ch Power

Carrier-1 -9.83 dBm / 4.515MHz

Offset Freq (MHz)	BW (MHz)		dBc / dBm		dBc / dBm
5.000 000	4.515 000	L1	-43.81 (-53.64)	U1	-44.50 (-54.33)
10.000 000	4.515 000	L2	-72.16 (-81.99)	U2	-72.26 (-82.09)

(a)



MKR 1 1.993 200 000 00 GHz -30.81 dBm RBW 30kHz ATT 10dB SWT 2.0s

Reference Level -15.00dBm RMS 1001 points

Center 1.992 50GHz Span 25.000 000MHz

Adjacent Channel Power (Carrier-1) - Offset Ch Power

Carrier-1 -9.89 dBm / 4.515MHz

Offset Freq (MHz)	BW (MHz)		dBc / dBm		dBc / dBm
5.000 000	4.515 000	L1	-66.86 (-76.75)	U1	-66.54 (-76.43)
10.000 000	4.515 000	L2	-72.75 (-82.64)	U2	-72.67 (-82.56)

(b)

Figure 4.20: ACLR comparison between OFDM and f-OFDM with Tx gain equal to 18 dB. (a) OFDM with no filter enabled. (b) 64 order FIR filter enabled.

high as 23.11 dB for a Tx gain of 18 dB (probably the most linear region of the Tx power amplifier). It can also be noticed that the leakage reduction for both filters orders is quite close. This means that the 64 order filter can be employed without any huge impact on the leakage onto adjacent channels while reducing the Tx flow processing time. On the other hand, the power leakage on adjacent channels with frequency offset of +/- 10 MHz is marginally decreased (at most 1.18 dB reduction), meaning that the leakage level on those channels is already very low even without filter. Figure 4.20 depicts one example of the ACLR measurement for a Tx gain of 18 dB with OFDM filtering disabled and enabled (64 order FIR

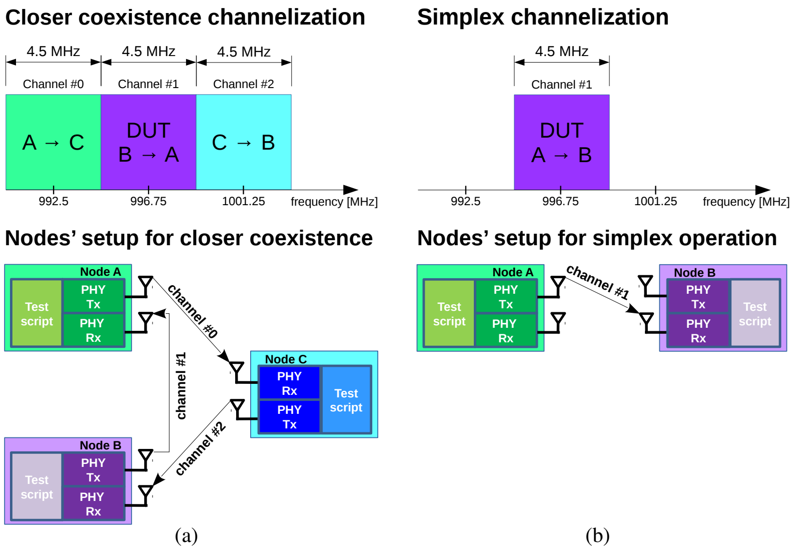


Figure 4.21: Closer coexistence experiments. (a) Channelization and nodes' setup for the closer coexistence experiment. (b) Channelization and nodes' setup for the simplex operation experiment.

filter) respectively. It is easily noticeable that the OFDM skirt is mitigated when the filter is enabled.

All the experiment results presented in this subsection show that the designed filters are able to attenuate out-of-band emissions without affecting in-channel performance.

4.8.2 Experiments with closer coexistence

In this section we present the results of experiments where we had a closer coexistence among nodes, i.e., the frequency offset between adjacent channels was made smaller. As known, the OFDM waveform is not well localized in frequency. Therefore, it needs guard-bands (GBs) at both ends of the transmitted BW to protect adjacent channels from spillage (also known as spectrum spread or leakage) of the OFDM signal into their bands. That is, the GBs are used so that the interference being inflicted on adjacent channels is reduced. For LTE systems, these GBs account for 10 % of the available BW and their purpose is to meet OOB emission level requirements. For example, for a PHY with BW of 10 MHz, GBs of 750 KHz are used, which accounts for 1.5 MHz in total of wasted spectrum. Therefore, the purpose of the experiments described in this section is to show that closer coexistence and consequently improved spectrum utilization, i.e., less spectrum wastage, is possible with well designed filters.

The setups used in the experiments are depicted in Figure 4.21. In Figure 4.21

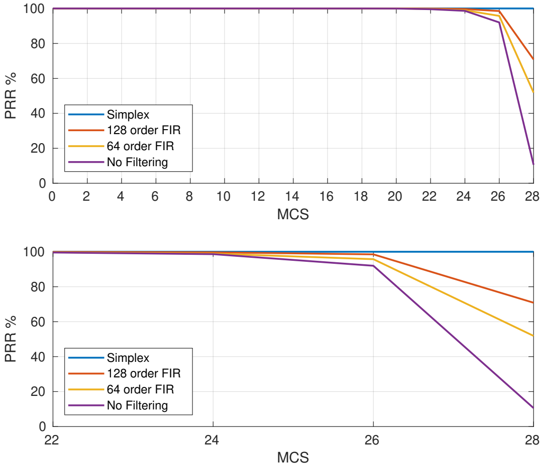


Figure 4.22: Packet Reception Rate for the experiment with no spacing between adjacent channels.

(a), we show the setup used to asses the closer coexistence. In this setup, as can be seen in the figure, three nodes simultaneously transmit data to each other on three different adjacent channels. Differently from standard LTE, where there would be a 10 % GB, the adjacent channels are now only 4.5 MHz apart from each other, which means that there is no GB at all between adjacent channels. The channel spacing of 4.5 MHz comes from the fact that the useful BW of a 5 MHz PHY is equal to 25 PRBs × 12 subcarriers per PRB × 15 KHz of subcarrier spacing. The closer coexistence experiment is split into three sub-experiments: (i) no filtering enabled in any of the three nodes, (ii) 64 order filters enabled in all the three nodes, and (iii) 128 order filters enabled in all the three nodes. For quantitative comparison, we used the simplex case, where one node transmits to the other on one given channel and there is no other transmissions on adjacent channels. This setup is shown in Figure 4.21 (b).

For all the experiments, Tx and Rx gains were set to 20 dB and COT of 10 ms, i.e., 10 slots of 1 ms with a gap of 0.5 ms between subsequent transmissions was used. We use 20 dB for Tx and Rx gains so that the received signal quality (i.e., signal-to-interference-plus-noise ratio (SINR)) would be good enough for the slot-based PHY to reach high MCS values. The gap of 0.5 ms is the lowest possible gap value with the current implementation of the slot-based PHY due to the latencies introduced by the 0MQ bus and the communication with the USRP. This gap is selected so that the channel occupancy time is higher than 95 % and with that, the interference to the device under test (DUT) will also happen more than 95 % of the time. A testbed with constant 60 dB path loss for all node pairs is used, meaning that the RF signals generated by one node experience 60 dB attenuation before reaching anyone of the other nodes. The testbed has a BW of 20 MHz centered at a frequency of 1 GHz, which means signals with frequencies less than 0.99 GHz

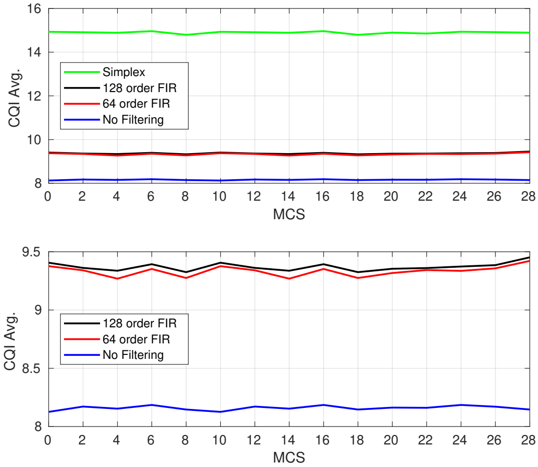


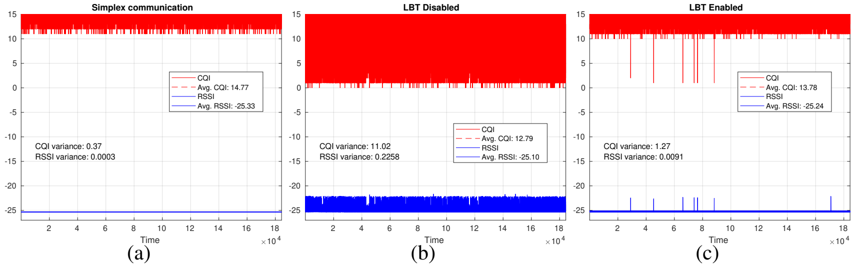
Figure 4.23: Averaged CQI for the experiment with no spacing between adjacent channels.

and/or greater than 1.01 GHz are heavily attenuated.

We quantitatively evaluate the closer coexistence and the impact of the filtering on the closer coexistence setup with two metrics, namely, PRR and average CQI, which are reported by the receiver. The PRR is calculated by counting the total number of detected slots (i.e., slots that had the PSS correlation peak higher than the predefined detection threshold) and the number of errors (i.e., the total number of synchronization and decoding errors). The results were collected during the transmission of 10000 frames with 10 1 ms-long slots for all MCS values (0 to 28).

Figures 4.22 and 4.23 present the PRR and average CQI results respectively. By analyzing figure 4.22, it becomes clear from observing the results that the interference is only noticeable for MCS values greater than 22, as the PRR is always equal to 100 % for MCS values ranging from 0 to 22. Therefore, the filtering will be only effective, and its provided improvement noticeable, for MCS values greater than 22. This is due to the fact that the FEC redundancy (i.e, Turbo coding) added to the transmitted data is high enough to make it much less susceptible to the interference caused by the leakage coming from the adjacent channels. As expected, in the closer coexistence sub-experiment with no filter enabled, the DUT suffers the highest interference as it has two radios simultaneously transmitting in the upper and lower adjacent channels. Additionally, when 64 order FIR filters are enabled in all nodes, the PRR increases from 10 % (case with no filtering) to approximately 52 % and for the case where the 128 order FIR filters are enabled in all the nodes, the PRR increases to more than 70 %.

In figure 4.23, average CQI results are presented. The top plot shows the result for all the cases, while the bottom plot excludes the average CQI for the Simplex case so that the results for the cases where filtering is enabled can be clearly analyzed. As can be seen, the average CQI increases from 8.1 to more than 9.3 when



(a) (b) (c)

Figure 4.24: Comparison when LBT is disabled and enabled. (a) Simplex operation: one radio transmits and the other one only receives. (b) Two radios transmitting on the same channel without any contention mechanism. (c) Two radios transmitting on the same channel with LBT enabled.

the 128 oder FIR filter is employed, meaning that the interference being caused by the spillage decreases, consequently increasing the SINR, which in turn increases the CQI. CQI is a quantized version of the SINR and indicates the highest MCS at which the block error rate (BLER) for the current channel does not exceed 10 % [83]. Moreover, it can be noticed that the average CQI for the 128 order FIR case is slightly higher than the one for the 64 order FIR filter. It is also observable that even with the filtering enabled, the resulting CQI is still lesser than that observed for the simplex case. It is explained by two facts: (i) the FIR filter orders employed here can not eliminate the OOB completely and some of it still spills over the adjacent channels. Besides that, the filter is added before the power amplifier (PA), which amplifies the reminiscent leakage. (ii) the filter's frequency response is not totally flat throughout the passband, especially at the edges, where it attenuates the upper and left parts of the transmitted signal. These two facts directly impact on the SINR, which in turn, impacts the CQI.

With these experiments, we show that when GBs are removed, the f-OFDM waveform achieves MCS 28 with PRR above 70 %, which is 60 % better than the case with no filter enabled. Therefore, with the right filtering and just enough channel spacing, the spectrum utilization can be improved.

4.8.3 Experiment with the LBT module

In this section we describe the results of an experiment with the LBT module. The main goal of this experiment is to show that LBT can decrease the number of collisions and consequently improve the coexistence among radios operating at the same frequency.

The experiment consists of three different steps. For all the three different steps we want to assess the coexistence performance in terms of CQI and RSSI measurements. The CQI is a quantized version of the SINR measurement that indicates the channel quality. The receiver reports CQI measurements back to the transmitter indicating the data rate supported by the channel at that instant. This

Table 4.8: LBT experiment measurements.

Step	CQI		RSSI		Collision
	Mean	Variance	Mean	Variance	%
Simplex	14.77	0.37	-25.33	0.0003	0
LBT Disabled	12.79	11.02	-25.10	0.2258	31.813
LBT Enabled	13.78	1.27	-25.24	0.0091	0.038

measurement report helps the transmitter to choose the optimum modulation and code rate, i.e., MCS, for its transmission. CQI is quantized as shown in [83]. The SINR is calculated as follows

$$\text{SINR} = 10\log_{10}\left(\frac{\text{RSRP}}{\text{noise\_power}_{\text{AVG}}}\right), \tag{4.8}$$

where Reference Signal Received Power (RSRP) is the linear average of reference signal power across the specified BW (in number of PRBs). When a collision happens, the average noise power increases as the interference being caused by other radios' transmissions is perceived by the receiver as effective noise. Consequently, the SINR decreases, which in turn decreases the CQI. The RSRP is calculated as

$$\text{RSRP} = \frac{\sum_{k=0}^{P-1} |Y_p(k)|^2}{P}, \tag{4.9}$$

where $Y_p(k)$ represents the received pilot symbol values in the frequency domain, i.e., the value of resource element (RE) carrying CRS over the entire PHY BW, and $P = 8 \ \times$ number of PRBs used by PHY. The average noise power over all REs carrying pilot symbols is estimated as [84, 85]

$$\text{noise\_power}_{\text{AVG}} = \frac{1}{P}\sum_{k=0}^{P-1} |Y_p(k) - \tilde{H}_p(k)X_p(k)|^2, \tag{4.10}$$

where $X_p(k)$ represents the known transmitted pilot symbol values and $\tilde{H}_p(k)$ is the estimated channel response for the RE occupied by the k-th pilot symbol. The channel response at REs carrying pilot symbols is obtained by dividing the received pilot symbols by their expected values,

$$\tilde{H}_p(k) = \frac{Y_p(k)}{X_p(k)} = H_p(k) + \text{noise}. \tag{4.11}$$

The other performance metric used in the experiment is the RSSI. It measures the average total received power observed over the whole slot (i.e., 1 ms) duration. The RSSI is calculated as

$$\text{RSSI} = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1} |y(n)|^2}{N}\right), \tag{4.12}$$

where $y(n)$ is the received slot signal in time domain, N is the slot size in number of complex samples, which is, for example, equal to 5760 samples for a 5 MHz PHY. When a collision happens, the transmitted signals (i.e., slots) overlap and consequently, the measured RSSI will be higher as the received signal is the combination of several transmissions that happen at the same time and frequency.

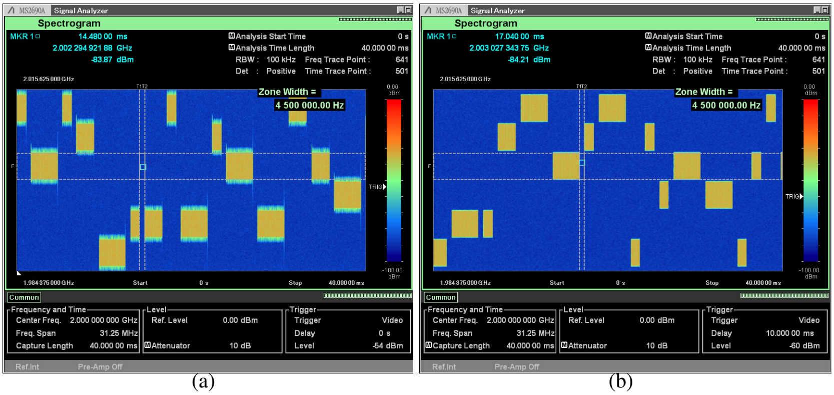
For all the three steps of this experiment we used a testbed that provides a constant 60 dB path loss for all radio pairs and a BW of 6 MHz centered at a frequency of 1 GHz, i.e., transmitted signals with BW wider than 6 MHz are heavily attenuated. The radios' Tx and Rx gains are set to 20 dB, COT is set to 25 ms, i.e., 25 slots are transmitted in sequence without any gap between subsequent slots, and the idle time is set to 10 ms, i.e., the interval between COTs. We transmitted more than 180×10^3 slots with MCS equal to 0 for each one of the steps of the experiment.

For the first step, we assess the performance when there is no collision involved. In this step there are two radios working in simplex mode, i.e., one radio transmits and the other one only receives. In this step there are no collisions. As can be seen in Figure 4.24 (a), RSSI is constant at around 25 dBW and CQI varies between 11 and 15 throughout the trial.

In the second step, we want to assess the performance when two radios compete for access to the medium (i.e., two radios operating in full-duplex mode at the same central frequency) without any contention mechanism. Figure 4.24 (b) shows that RSSI is not constant anymore, varying between -25 and -21 dBW while CQI varies between 0 and 15 throughout the trial. The fluctuation in both CQI and RSSI values are due to collisions. As a way to estimate the number of collisions during the step, we count as a collision CQI measurements with value less than 11, which is the minimum CQI value for the simplex communication case, where no collisions happen. By using this approach, the percentage of collisions is equal to 31.813 $\%$.

The third step is similar to the second step with the only difference being the use of a contention mechanism, i.e., LBT. In this step we assess the performance when LBT is enabled with a threshold of -70 dBW, CCA of 173.26 μs and a random backoff of at most 32 CCA intervals. Figure 4.24 (c) shows that RSSI is much more constant, staying around -25 dBW with very few points where it goes to -21 dBW. The CQI behavior is also closer to that depicted in (a), with CQI varying between 10 and 15 and very few points where it drops to 0 throughout the trial. The percentage of collisions is equal to 0.038 $\%$ using the same approach described earlier to estimate the number of collisions. Table 4.8 summarizes the results of the three steps of the experiment.

Based on the results of this experiment, it is clearly seen that either CQI or RSSI measurements (or even both of them) could be used as an approach to detect collisions. For example, these measurements could be employed in LTE systems to detect collisions more effectively in a timely manner and consequently, decrease the inherent latency of those systems. In LTE systems, due to the inherent latencies introduced by the LTE protocol stack, the HARQ feedback associated to a certain subframe is received at least 4 ms after its transmission time [86].



(a)

MKR 1 □ 14.480 00 ms
2.002 294 921 88 GHz
-83.87 dBm

Analysis Start Time 0 s
Analysis Time Length 40.000 00 ms
RBW : 100 kHz Freq Trace Point : 641
Det : Positive Time Trace Point : 501

Frequency and Time
Center Freq. 2.000 000 000 GHz
Freq. Span 31.25 MHz
Capture Length 40.000 00 ms

Level
Ref. Level 0.00 dBm
Attenuator 10 dB

Trigger
Trigger Video
Delay 0 s
Level -54 dBm

(b)

MKR 1 □ 17.040 00 ms
2.003 027 343 75 GHz
-84.21 dBm

Analysis Start Time 0 s
Analysis Time Length 40.000 00 ms
RBW : 100 kHz Freq Trace Point : 641
Det : Positive Time Trace Point : 501

Frequency and Time
Center Freq. 2.000 000 000 GHz
Freq. Span 31.25 MHz
Capture Length 40.000 00 ms

Level
Ref. Level 0.00 dBm
Attenuator 10 dB

Trigger
Trigger Video
Delay 10.000 00 ms
Level -60 dBm

Figure 4.25: Comparison of MF-TDMA feature without and with filtering. (a) OFDM without filtering. (b) 128 order FIR filter enabled.

4.8.4 Experiment with MF-TDMA feature

In this section we show the capability of the slot-based PHY to work in MF-TDMA mode with the use of the transmission timestamp field in the Tx control message. MF-TDMA is a two-dimensional multiple access scheme that combines frequency division with time division access to the medium. Due to its efficiency and flexibility, MF-TDMA scheme has been widely used in several communications systems such as very small aperture terminal (VSAT) satellite communications systems [87]. A MF-TDMA communications system is composed of several channels, where each one of the channels is divided into a number of time slots. Such scheme ensures adaptability and improves the radio resources utilization for multi-user and multi-service applications [87].

Figure 4.25 shows two spectrograms collected during 40 ms over a 31.25 MHz BW (the Anritsu MS2690A Signal Analyzer does not allow a BW of 30 MHz, being 31.25 MHz the closest configurable value) with the slot-based PHY set to operate with six 5 MHz channels (i.e., 25 RBs) with a channel spacing of 2.5 MHz. In this experiment, one radio transmits a random number of slots at randomly selected channels. The channel number and number of transmitted slots, i.e., COT, are randomly selected between the ranges 0-5 and 1-3 respectively.

Figure 4.25 (a) presents the spectrogram for the case with no filtering enabled (i.e., OFDM waveform) and with a gap of 0.5 ms between consecutive transmissions. As can be noticed, with the current channel spacing, the OFDM OOB emissions might cause interference to adjacent channels and consequently decrease the system's throughput. The OOB emissions can be mitigated with the use of filtering as shown next.

Figure 4.25 (b) depicts the spectrogram for the case when the f-OFDM waveform is enabled with a gap of 0.5 ms between consecutive transmissions. As can be seen, when the f-OFDM is used, the OOB emissions are mitigated and conse-

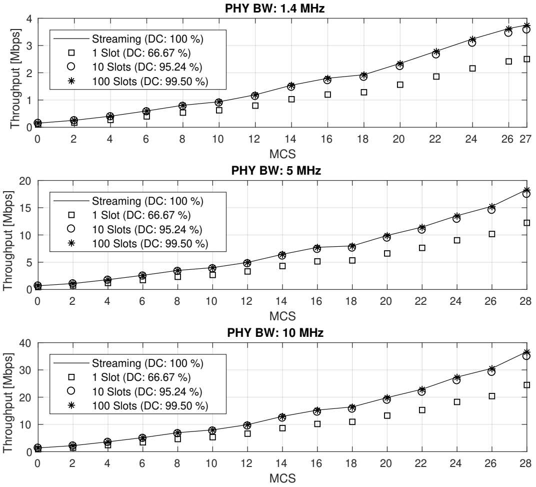


Figure 4.26: Slot-based PHY throughput for 1.4, 5 and 10 MHz PHY BWs, different MCS/duty cycle values and 128-th order FIR filtering.

quently, interference to and from adjacent channels is also mitigated. Additionally, another consequence of the filtering use is that the channel spacing could be made smaller and consequently improve the radio resource utilization performance, i.e., decrease the spectrum wastage.

Figure 4.26 depicts throughput measurements taken with the slot-based PHY for 1.4, 5 and 10 MHz PHY BWs, several MCS and duty cycle values working in full-duplex mode (i.e., the slot-based PHY is simultaneously transmitting and receiving). We use full-duplex mode in order to check if it impacts somehow the measured throughout once in full-duplex mode the slot-based PHY is being fully utilized. The measurements were taken for COTs of 1, 10 and 100 ms with a gap of 0.5 ms between subsequent transmissions. This means that the duty cycle increases as the number of slots sent in a row also increases. The throughput was averaged over 10 measurement intervals of 10 seconds each. During one measurement interval (i.e., 10 s) the number of received bits is counted and then divided by the interval to produce the throughput measured during that interval. For this experiment, we have also enabled 128-th order FIR filtering. In the figure, for comparison reasons, we also add the theoretical maximum throughput achieved by the *Streaming* mode, where there is no gap between subsequent transmissions, i.e., a duty cycle equal to 100 %. The theoretical maximum throughput is calculated diving the TB size in bits for each MCS by 1 ms. As expected, the measured slot-based PHY's throughput approaches the theoretical maximum throughput for all MCS and PHY BW values as the duty cycle increases, yielding more than 36.5 Mbps for a PHY BW of 10 MHz and MCS 28. Additionally, the operation in full-duplex mode has no visible impact on the achieved throughput. This is due to the powerful server, with 12 cores, used to run the slot-based PHY.

Table 4.9: Average Tx and Rx processing times for different PHY BWs and MCS values with 64 order FIR filter.

PHY BW: 1.4 MHz slot size: 1920 I/Q samples						
MCS	Tx processing time [ms]			Rx processing time [ms]		
	Mod.	Filter	Total	Synch.	Demod.	Total
0 (QPSK)	0.030	0.049	0.079	0.051	0.114	0.165
10 (16QAM)	0.052	0.047	0.099	0.052	0.124	0.176
18 (64QAM)	0.053	0.048	0.101	0.049	0.140	0.189
27 (64QAM)	0.065	0.047	0.112	0.046	0.196	0.242

PHY BW: 5 MHz slot size: 5760 I/Q samples						
MCS	Tx processing time [ms]			Rx processing time [ms]		
	Mod.	Filter	Total	Synch.	Demod.	Total
0 (QPSK)	0.072	0.120	0.192	0.185	0.166	0.351
10 (16QAM)	0.101	0.122	0.223	0.185	0.212	0.397
18 (64QAM)	0.135	0.120	0.255	0.187	0.284	0.471
28 (64QAM)	0.206	0.121	0.327	0.186	0.342	0.528

PHY BW: 10 MHz slot size: 11520 I/Q samples						
MCS	Tx processing time [ms]			Rx processing time [ms]		
	Mod.	Filter	Total	Synch.	Demod.	Total
0 (QPSK)	0.134	0.285	0.419	0.383	0.283	0.666
10 (16QAM)	0.175	0.283	0.458	0.384	0.407	0.791
18 (64QAM)	0.256	0.287	0.543	0.382	0.543	0.925
28 (64QAM)	0.394	0.286	0.680	0.385	0.655	1.040

4.8.5 slot-based PHY Profiling

In this section we present some measurements regarding the average Tx and Rx processing times of the proposed slot-based PHY. Table 4.9 shows the average processing times for PHY BWs of 1.4, 5 and 10 MHz for MCS values of 0, 10, 18, 27 and 28 which use QPSK, 16QAM and 64QAM modulations as indicated in the table. With a PHY BW of 1.4 MHz it is not possible to reach the highest MCS value, i.e., 28, due to the fact that the slot structure uses two OFDM symbols to carry control information and another two OFDM symbols to carry PSS and SSS signals (see Figure 4.5), which makes the effective payload size (i.e., the number of data bits an 1 ms long slot can transport) smaller than the transport block size.

The processing times were averaged over 10000 transmissions with COT set to 1 ms, i.e., only 1 slot for each new transmission and a gap of 0.5 ms between subsequent transmissions. For these measurements we set both Tx and Rx gains

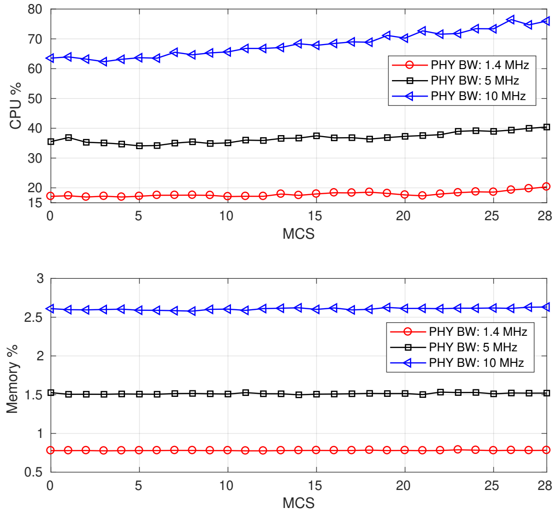


Figure 4.27: CPU and memory utilization of the slot-based PHY.

to 20 dB so that the received signal quality (i.e., SINR) would be good enough for the PHY to reach high MCS values.

The Modulation, *Mod.*, column lists the average time it takes to process control/data bits (i.e., channel coding, rate matching, scrambling, modulation, precoding and resource grid mapping) and generate a 1 ms long slot. The Filter column shows the average filtering time. As can be seen, for a given PHY BW, the filtering time is approximately constant. It shows that the filtering time does not depend on the MCS being used as, for a given PHY BW, a slot always has the same size irrespective of the MCS. The Synchronization, *Synch.*, column shows the average time it takes to correlate the received signal with the local PSS sequence, decode the SSS signal, estimate CFO, align the subframe to the buffer start (read from the USRP missing samples necessary to have a full slot in the buffer), and correct CFO. As seen from the table, the synchronization time is approximately constant for a given PHY BW, showing, as expected, that it does not depend on the MCS value. The synchronization time depends only on slot size, i.e., number of I/Q samples making up a slot. The Demodulation, *Demod.*, column lists the average time it takes for the already synchronized slot to be demodulated. As expected, the higher the MCS the higher the decoding time. That is due to the fact that the the information (data) carried by one slot is higher for high MCS values, and therefore, the processing time is longer. Another point that is also worth mentioning is that the demodulation process is the most time-consuming task carried out by PHY.

Next, we use the *psutil* library to assess CPU and memory utilization of the slot-based PHY. *psutil* is a Python cross-platform library for retrieving information on running processes and system utilization, such as CPU, memory, disks, network, sensors, etc. [88]. Our goal with this experiment is simply to evaluate the

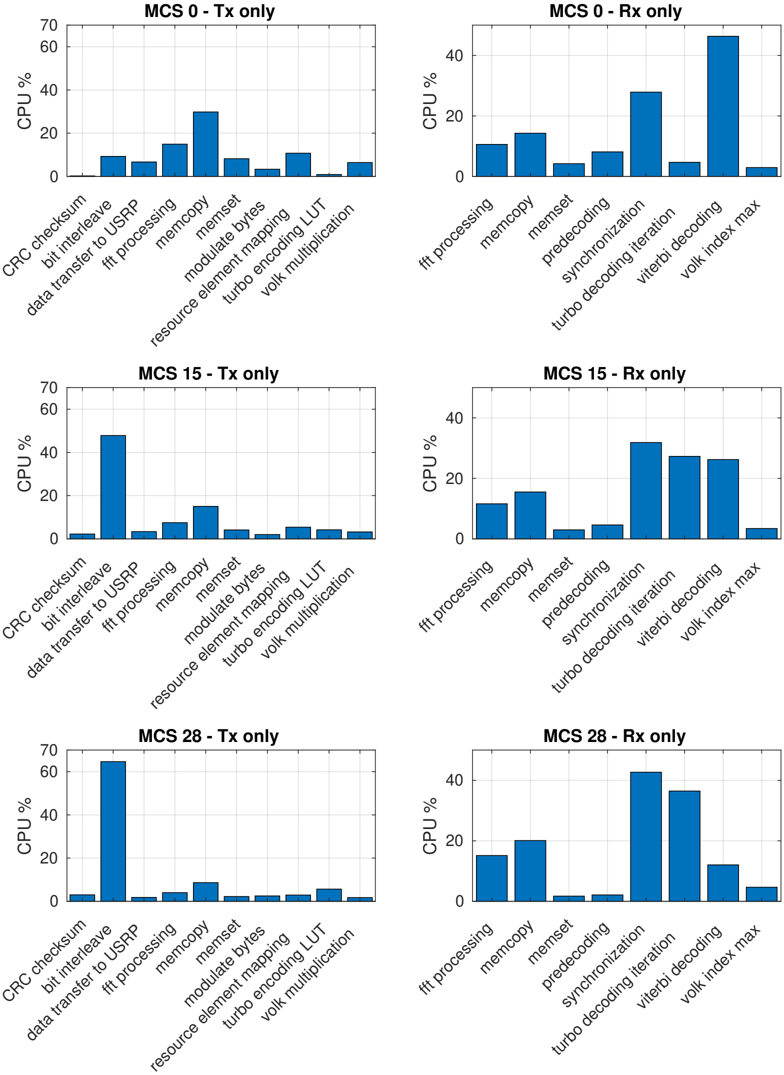


Figure 4.28: CPU profiling of individual components of the slot-based PHY when using a 5 MHz PHY BW.

computational expense of the slot-based PHY operation in full-duplex mode, i.e., simultaneously transmitting and receiving data. Therefore, we use a simple script connected to PHY that only transmits and receives data without any implementation of upper layer protocols. The CPU percentage scale used by the *psutil* lib is 100 % when a core is fully utilized or when one core is at 20 % load and a second one is at 80 % load. Therefore, with multicore CPUs the CPU utilization percentage can exceed 100 %. For instance, the *psutil* lib can return a CPU utilization from 0 % (idle application) up to 1200 % (application fully using the cores) for CPU with 12 cores.

In this experiment 2 nodes transmit 10×10^3 1 ms-long slots with a gap of 0.5 ms between subsequent transmissions to each other on two different channels (i.e., non-overlapping frequencies) and with Tx and Rx gains of both nodes set to 20 dB. We used a testbed that provides constant 60 dB attenuation between the radios and a BW of 6 MHz centered at a frequency of 1 GHz.

Figure 4.27 shows the CPU and memory usage of the slot-based PHY, operating at 1.4, 5 and 10 MHz BWs over the whole set MCS values. The results presented in the figure were obtained by averaging CPU and memory usage values sampled every 200 ms during the whole duration of the experiment. We sample CPU and memory utilization of only one of the nodes.

As expected, as the PHY BW increases, both CPU and memory utilization increase, however, there is no CPU or memory starvation issues with any of the PHY BW and MCS values. The CPU usage increases as the MCS increases. This is due to the fact that for higher MCS values (i.e., higher data rates), both Quadratic Permutation Polynomial (QPP) interleaver (which is part of the turbo encoding process at Tx side), synchronization and turbo decoding (both at Rx side) tasks gain weight and consume quite a lot of CPU for data processing. The CPU time spent by individual PHY functions is presented next and those results are in line with the results in [29, 89]. For a PHY BW of 10 MHz the CPU utilization is of around 75 % meaning that the processing power of only one core is being used, however, it is not being fully utilized, and most probably, leaving the other cores in idle state for large periods of time.

In the case of memory usage, it can be observed by analyzing the figure that for a given PHY BW it is practically constant for all MCS values and therefore, independent of the configured MCS value. This is the expected result as all the memory used is pre-allocated during the initialization of the slot-based PHY, i.e., there is no memory allocation during PHY operation. Based on these results, we conclude that the slot-based PHY does not exhaust CPU or memory resources as PHY BW or MCS increases.

In the following, we evaluate the CPU consumption of independent processing tasks (i.e., functions) within the slot-based PHY software. For this evaluation we use *valgrind* with its *callgrind* tool. *Callgrind* is a profiling tool that records the call history among functions in a program's run as a call-graph through the use of runtime instrumentation [90]. By default, the collected data consists of the number of instructions executed, their relationship to source lines, the caller/callee relationship between functions, and the numbers of such calls [90].

Valgrind adds a considerable amount of instrumentation code to the application it is profiling, which makes the application 4 to 50 times slower [91]. Therefore, when running the slot-based PHY through *valgrind* with the *calgrind* tool, we noticed that when transmitting slots, PHY was not being able to feed samples at the USRP's expected sample rate and in consequence, large parts of a slot were not being transmitted due to underflows. For that reason, in order to be able to profile PHY, we decided to split the profiling into two parts, one where only the Tx thread is spawn (i.e., all Rx threads are disabled) and a second part where only the Rx threads are spawn (i.e., the Tx thread is disabled).

The setup for this experiment is the same as the one used earlier for CPU and memory profiling. In the first part, where only the Tx thread is started, the node exclusively transmits slots while PHY is profiled. For the first part, as we are only interested in the profiling of Tx related functions, there is no need for a second node receiving the data. In the second part, where only the Rx threads are started, the node with PHY being profiled receives data from a second node, which is not being profiled.

Our findings are summarized in Figure 4.28 for a PHY with BW equal to 5 MHz and different MCS values. The figure depicts the most representative CPU consumers, i.e., functions that use more CPU processing for the two considered parts of the experiment. The left column shows CPU time spent by functions related to Tx processing, while the right column presents CPU time spent by functions related to Rx processing.

As can be seen, for Tx related functions, bit interleaving (i.e., QPP interleaver) gains weight as MCS increases, consuming approximately 65 % of CPU time for MCS equal to 28 compared to only 9 % when MCS is equal to 0. For Rx related functions, it is worth highlighting how the synchronization task consumes a great deal of CPU time for all the MCS values considered in this experiment ranging from 28 % to more than 42 % of CPU time. It is also worth mentioning how the turbo decoding task gains weight as MCS increases, going from 5 % to more than 36 % for MCS values equal to 0 and 28 respectively. It is also noticeable that the viterbi decoding weight decreases with the MCS, starting at 46 % for MCS 0 and dropping to 12 % when MCS is equal to 28.

Regarding the large CPU % taken by the synchronization processing as MCS increases. We don't have a definitive answer to that, but we have a theory to explain that. Functions like synchronization, turbo decoding, FFT, and memcpy (all of them have increased CPU consumption as MCS increases) share the limited size of the CPU's cache. For high MCS, the greater number of LLR/bits (turbo decoder) occupies a larger portion of the cache, and then, fewer I/Q samples can stay in cache. Due to this, cache miss (swap in/out) would happen more frequently, and the synchronization function, which relies on many I/Q samples, might need to wait for I/Q samples coming from memory to cache. That increases the execution time, which could be related to the higher CPU consumption.

4.9 Conclusions

This chapter presented a open source SDR-based framework that enables experimental research and prototyping for various next generation wireless networks spectrum sharing scenarios. Its high configurability, supported by an interface built upon popular programming libraries (Google protobuf and ZeroMQ), allows engineers and researchers to easily modify it or extend its functions by plugging in new modules implementing novel spectrum sharing techniques and approaches. Additionally, the proposed slot-based PHY is of great importance to combat spectrum scarcity as it has a pivotal role in providing optimum utilization of time-frequency resources. These features make it the perfect candidate for an extensive range of spectrum sharing experimentations in real-world or realistic environments such as testbeds aiming at better understanding disruptive spectrum sharing schemes. Three distinctive use cases are presented with hints on how the proposed framework could be employed in intelligent spectrum sharing research.

Acknowledgment

This work was funded by the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 732174 – Orchestration and Reconfiguration Control Architecture (ORCA) project.

References

[1] Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020*, White paper, February 2016.

[2] The 5G Infrastructure Public Private Partnership (5G-PPP) Technical Report, *5G empowering vertical industries*, February 2016.

[3] 3GPP Technical Report TR 38.913, *Study on scenarios and requirements for next generation access technologies, version 14.2.0, Release 14*, March 2017.

[4] Michael Koziol, *5G New Radio and What Comes Next*, IEEE Spectrum, January 2018.

[5] Gregory Staple and Kevin Werbach, *The End of Spectrum Scarcity*, IEEE Spectrum, March 2014.

[6] Recommendation ITU-R M.2083, *IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond*, September 2015.

[7] ITU-R Report M.2370-0, *IMT traffic estimates for the years 2020 to 2030*, July 2015.

[8] Keith Mallinson, *The path to 5G: as much evolution as revolution*, May 2016. [available at] http://www.3gpp.org/news-events/3gpp-news/1774-5g_-wiseharbour.

[9] Oumer Teyeb, Gustav Wikstrom, Magnus Stattin, Thomas Cheng, Sebastian Faxer, and Hieu Do, *Evolving LTE to fit the 5G future*, Ericsson Technology Review, January 2017. [available at] https://www.ericsson.com/en/ericsson-technology-review/archive/2017/evolving-lte-to-fit-the-5g-future

[10] Konstantinos Chatzikokolakis, Panagiotis Spapis, Alexandros Kaloxylos, and Nancy Alonistioti, *Toward spectrum sharing: opportunities and technical enablers*, IEEE Communications Magazine, vol. 53, no. 7, July 2015.

[11] Junyu Liu, Min Sheng, Lei Liu, and Jiandong Li, *Network Densification in 5G: From the Short-Range Communications Perspective*, IEEE Communications Magazine, vol. 55, no. 12, December 2017.

[12] Yan Li, and Shaoyi Xu, *Traffic Offloading in Unlicensed Spectrum for 5G Cellular Network: A Two-Layer Game Approach*, MDPI Entropy, January 2018.

[13] Federal Communications Commission, *Spectrum Policy Task Force Report*, [online]. Available: https://www.fcc.gov/document/spectrum-policy-task-force, FCC, November 2002.

[14] Mina Labib, Vuk Marojevic, Jeffrey H. Reed, and Amir I. Zaghloul, *Extending LTE into the Unlicensed Spectrum: Technical Analysis of the Proposed Variants*, IEEE Communications Standards Magazine, vol. 1, no. 4, December 2017.

[15] Qualcomm Technologies, Inc., *LTE-U/LAA, MuLTEfire and Wi-Fi; making best use of unlicensed spectrum*, White Paper, September 2015.

[16] MulteFire Alliance, *MulteFire Release 1.0 Technical Paper: A new way to wireless*, January 2017.

[17] Andreas Roessler, *Impact of spectrum sharing on 4G and 5G standards a review of how coexistence and spectrum sharing is shaping 3GPP standards*, IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), October 2017.

[18] Ahsan Saadat, Gengfa Fang, and Wei Ni, *A Two-Tier Evolutionary Game Theoretic Approach to Dynamic Spectrum Sharing through Licensed Shared Access*, IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, October 2015.

[19] Nadine Abbas, Youssef Nasser, and Karim El Ahmad, *Recent advances on artificial intelligence and learning techniques in cognitive radio networks*, EURASIP Journal on Wireless Communications and Networking, vol. 2015, no. 1, pp. 174-194, December 2015.

[20] Seppo Yrjola, and Heikki Kokkinen, *Licensed Shared Access evolution enables early access to 5G spectrum and novel use cases*, EAI Endorsed Transactions on Wireless Spectrum, December 2017.

[21] J. Mitola, *The software radio architecture*, IEEE Communications Magazine, vol. 33, no. 5, May 1995.

[22] J. Mitola, *Software radios-survey, critical evaluation and future directions*, National Telesystems Conference (NTC), May 1992.

[23] Felipe A. P. de Figueiredo, *A Framework for Intelligent Spectrum Sensing and Sharing*, [online]. Available: https://github.com/zz4fap/intelligent-spectrum-sharing, May 2018.

[24] Felipe A. P. de Figueiredo,Xianjun Jiao, Wei Liu, Irfan Jabandzic, Spilios Giannoulis, and Ingrid Moerman, *A Framework for Intelligent Spectrum Sharing*, Preprints 2018, 2018080447 (doi: 10.20944/preprints201808.0447.v1).

[25] Ettus Research LLC, *USRP Hardware Driver*, [online]. Available: http://files.ettus.com/manual/

[26] Ettus Research LLC, *Products*, [online]. Available: https://www.ettus.com/product

[27] *Open Air Interface*, 2015. [online]. Available: http://www.openairinterface.org/

[28] *openLTE*, 2015. [online]. Available: http://openlte.sourceforge.net/

[29] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano and D. J. Leith, *srsLTE: An Open-Source Platform for LTE Evolution and Experimentation*, ACM WiNTECH Workshop, October 2016.

[30] Amarisoft Corp. *Amari LTE 100 - Software LTE base station on PC*, [online]. Available: http://www.amarisoft.com

[31] National Instruments, *LabVIEW Communications LTE Application Framework 2.0 and 2.0.1*, White Paper, December 2016. [online]. Available: http://www.ni.com/white-paper/53286/en/

[32] National Instruments, *Real-time LTE/Wi-Fi Coexistence Testbed*, White Paper, February 2016. [online]. Available: http://www.ni.com/white-paper/53044/en/

[33] Zhenyu Zhou, Yunjian Jia, Fei Chen, Kimfung Tsang, Guangyi Liu, and Zhu Han, *Unlicensed Spectrum Sharing: From Coexistence to Convergence*, IEEE Wireless Communications, vol. 24, no. 5, October 2017.

[34] Cristina Cano, Douglas J. Leith, Andres Garcia-Saavedra, and Pablo Serrano, *Fair Coexistence of Scheduled and Random Access Wireless Networks: Unlicensed LTE/Wi-Fi*, IEEE/ACM Transactions on Networking, vol. 25, no. 6, December 2017.

[35] D. Ruby, M. Vijayalakshmi, and A. Kannan, *Intelligent relay selection and spectrum sharing techniques for cognitive radio networks*, Cluster Computing, August 2017.

[36] Usama Mir, Leila Merghem-Boulahia, Dominique Gaiti, *Dynamic Spectrum Sharing in Cognitive Radio Networks: a Solution based on Multiagent Systems*, International Journal on Advances in Telecommunications, vol. 3 no. 3, January 2010.

[37] Nadisanka Rupasinghe, and Ismail Guvenc, *Reinforcement learning for licensed-assisted access of LTE in the unlicensed spectrum*, IEEE Wireless Communications and Networking Conference (WCNC), June 2015.

[38] Ruben Mennes, Miguel Camelo, Maxim Claeys and Steven Latre, *A Neural-Network-based MF-TDMA MAC Scheduler for Collaborative Wireless Networks*, IEEE Wireless Communications and Networking Conference (WCNC), April 2018.

[39] Zoraze Ali, Lorenza Giupponi, Josep Mangues-Bafalluy, and Biljana Bojovic, *Machine learning based scheme for contention window size adaptation in LTE-LAA*, IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), February 2018.

[40] P. Hintjens, *ZeroMQ Messaging for Many Applications*, O'Reilly Media, March 2013.

[41] Ettus Research LLC, *USRP X Series*, [online]. Available: https://www.ettus.com/product/category/USRP-X-Series

[42] National Instruments, *Overview of the NI USRP RIO Software Defined Radio*, White Paper, December 2015. [online]. Available: http://www.ni.com/white-paper/52119/en/

[43] Google, *protobuf - Protocol Buffers - Google's data interchange format*, [online]. Available: http://code.google.com/p/protobuf/, 2011.

[44] DARPA SC2 Website: https://spectrumcollaborationchallenge.com/.

[45] DARPA, *Kick-off meeting*, The Johns Hopkins University, January 2017.

[46] Cristina Cano, and Douglas J. Leith, *Unlicensed LTE/Wi-Fi coexistence: Is LBT inherently fairer than CSAT?*, IEEE International Conference on Communications (ICC), July 2016.

[47] Houman Zarrinkoub, *Overview of the LTE Physical Layer*, John Wiley & Sons, January 2014.

[48] Ali A. Zaidi, Robert Baldemair, Hugo Tullberg, Hakan Bjorkegren, Lars Sundström, Jonas Medbo, Caner Kilinc, and Icaro Da Silva, *Waveform and Numerology to Support 5G Services and Requirements*, IEEE Communications Magazine, vol. 54, no. 11, pp. 90-98, November 2016.

[49] F. A. P. de Figueiredo, N. F. T. Aniceto, J. Seki, I. Moerman and G. Fraidenraich, "Comparing f-OFDM and OFDM Performance for MIMO Systems Considering a 5G Scenario," 2019 IEEE 2nd 5G World Forum (5GWF), 2019, pp. 532-535, doi: 10.1109/5GWF.2019.8911702.

[50] Sravanthi Kanchi, Shubhrika Sandilya, Deesha Bhosale, Adwait Pitkar, and Mayur Gondhalekar, *Overview of LTE-A technology*, IEEE Global High Tech Congress on Electronics (GHTCE), March 2014.

[51] Shao-Yu Lien, Shin-Lin Shieh, Yenming Huang, Borching Su, Yung-Lin Hsu, and Hung-Yu Wei, *5G New Radio: Waveform, Frame Structure, Multiple Access, and Initial Access*, IEEE Communications Magazine, vol. 55, no. 6, June 2017.

[52] Naga Bhushan, Tingfang Ji, Ozge Koymen, John Smee, Joseph Soriaga, Sundar Subramanian, and Yongbin Wei, *5G Air Interface System Design Principles*, IEEE Wireless Communications, vol. 24, no. 5, pp. 6-8, October 2017.

[53] ANSI/IEEE 802.11n-2009 , *IEEE Standard for Information technology - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, 2010.

[54] ETSI TR 101.190, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television - V1.6.1*, January 2009.

[55] Feng Wang, and Yu Zhu, *An efficient CFO estimation algorithm for the downlink of 3GPP-LTE*, International Conference on Wireless Communications and Signal Processing (WCSP), November 2011.

[56] Gwanmo Ku, and John MacLaren Walsh, *Resource Allocation and Link Adaptation in LTE and LTE Advanced: A Tutorial*, IEEE Communications Surveys & Tutorials, vol. 17, no. 3, December 2014.

[57] 3GPP TS 36.213, *Physical Layer Procedures*, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA). URL: http://www.3gpp.org, May 2016.

[58] Quentin Bodinier, Faouzi Bader, and Jacques Palicot, *On Spectral Coexistence of CP-OFDM and FB-MC Waveforms in 5G Networks*, IEEE Access, vol. 5, July 2017.

[59] Hao Lin, and Pierre Siohan, *Major 5G Waveform Candidates: Overview and Comparison*, Book chapter in Signal Processing for 5G, Wiley, August 2016.

[60] Lei Zhang, Ayesha Ijaz, Pei Xiao, Mehdi M. Molu, and Rahim Tafazolli, emphFiltered OFDM Systems, Algorithms, and Performance Analysis for 5G and Beyond, IEEE Transactions on Communications, vol. 66, no. 3, March 2018.

[61] Javad Abdoli, Ming Jia, and Jianglei Ma, *Filtered OFDM: A New Waveform for Future Wireless Systems*, IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), August 2015.

[62] Xi Zhang, Ming Jia, Lei Chen, Jianglei Ma, and Jing Qiu, *Filtered-OFDM - Enabler for Flexible Waveform in the 5th Generation Cellular Networks*, IEEE Global Communications Conference (GLOBECOM), December 2015.

[63] Felipe A. P. de Figueiredo, Fabbryccio A. C. M. Cardoso, Jose A. Bianco F., Rafael M. Vilela, Karlo G. Lenzi, *Multi-stage Based Cross-Correlation Peak Detection for LTE Random Access Preambles*, Revista Telecomunicacoes, vol. 15, September 2013.

[64] Janne J. Lehtomaki, Johanna Vartiainen, Markku Juntti and Harri Saarnisaari, *Spectrum sensing with forward methods*, in Proc. IEEE Military Commun. Conf., pp. 1-7, Washington, D.C., USA, October 2006.

[65] Gen Li, Tan Wang, Qingyu Miao, Ying Wang, and Biao Huang, *Spectrum Sharing for 5G*, 5G Mobile Communications, Springer, Cham, October 2016.

[66] X. Wang et al., *Throughput and Fairness Analysis of Wi-Fi and LTE-U in Unlicensed Band*, IEEE JSAC, vol. 35, no. 1, pp. 63-78, January 2017.

[67] 3GPP TR 36.889, *Feasibility Study on Licensed-Assisted Access to Unlicensed Spectrum (Release 13)*, January 2015.

[68] Mina Labib, Vuk Marojevic, Jeffrey H. Reed, and Amir I. Zaghloul, *Extending LTE into the Unlicensed Spectrum: Technical Analysis of the Proposed Variants*, IEEE Communications Standards Magazine, vol. 1, no. 4, December 2017.

[69] Jaehong Yi, Weiping Sun, Seungkeun Park, and Sunghyun Choi, *Performance Analysis of LTE-LAA Network*, IEEE Communications Letters, December 2017.

[70] Xianjun Jiao, Ingrid Moerman, Wei Liu, and Felipe A. P. de Figueiredo, *Radio Hardware Virtualization for Coping with Dynamic Heterogeneous Wireless Environments*, International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM), September 2017.

[71] Felipe A. P. de Figueiredo, Xianjun Jiao, Wei Liu, and Ingrid Moerman, *Radio Hardware Virtualization for Software-Defined Wireless Networks*, Wireless Personal Communications, vol. 100, no. 1, pp. 113-126, March 2018.

[72] Vasilis Maglogiannis, Dries Naudts, Adnan Shahid, and Ingrid Moerman, *An adaptive LTE listen-before-talk scheme towards a fair coexistence with Wi-Fi in unlicensed spectrum*, Telecommunication Systems, January 2018.

[73] Anand M. Baswade, and Bheemarjuna Reddy Tamma, *Channel sensing based dynamic adjustment of contention window in LAA-LTE networks*, International Conference on Communication Systems and Networks (COMSNETS), March 2016.

[74] Qualcomm Technologies, Inc., *What can we do with 5G NR Spectrum Sharing that isn't possible today?*, White Paper, December 2017.

[75] Ericsson AB, *Spectrum sharing*, White Paper, October 2013.

[76] Hai-Yan Shi, Wan-Liang Wang, Ngai-Ming Kwok, and Sheng-Yong Chen, *Game Theory for Wireless Sensor Networks: A Survey*, MDPI Sensors, vol. 12, no. 7, July 2012.

[77] Saed Alrabaee, Mahmoud Khasawneh, Anjali Agarwal, Nishith Goel, and Marzia Zaman, *A game theory approach: Dynamic behaviours for spectrum management in cognitive radio network*, IEEE Globecom Workshops (GC Wkshps), December 2012.

[78] Prabodini Semasinghe, Setareh Maghsudi, and Ekram Hossain, *Game Theoretic Mechanisms for Resource Management in Massive Wireless IoT Systems*, IEE Communications Magazine, vol. 55, no. 2, February 2017.

[79] Ettus Research LLC, *CBX 1200-6000 MHz Rx/Tx (120 MHz, X Series only)*, [online]. Available: https://www.ettus.com/product/details/CBX120

[80] 3GPP TS 36.104, *Base Station (BS) radio transmission and reception - V14.3.0*, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA). URL: http://www.3gpp.org, April 2017.

[81] 3GPP TS 36.141, *Base Station (BS) Conformance Testing - V8.4.0*, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA). URL: http://www.3gpp.org, September 2009.

[82] 3GPP TS 36.521-1, *User Equipment (UE) Conformance Specification; Radio Transmission and Reception Part 1: Conformance Testing - V8.3.1*, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA). URL: http://www.3gpp.org, September 2009.

[83] Mohammad T. Kawser, Nafiz Imtiaz Bin Hamid, Md. Nayeemul Hasan, M. Shah Alam, and M. Musfiqur Rahman, *Downlink SNR to CQI Mapping for Different Multiple Antenna Techniques in LTE*, International Journal of Information and Electronics Engineering, vol. 2, no. 5, September 2012.

[84] 3GPP TS 36.141. *Base Station (BS) conformance testing*, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA). URL: http://www.3gpp.org.

[85] Van de Beek, J. J., O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, *On Channel Estimation in OFDM Systems*, IEEE Vehicular Technology Conference, vol. 2, 1995.

[86] Kazuki Takeda, Li Hui Wang, and Satoshi Nagata, *Latency reduction towards 5G*, IEEE Wireless Communications, vol. 24, no. 3, pp. 2-4, June 2017.

[87] D. Qijia, Z. Jun, Z. Tao, and Q. Yong, *Resource Allocation Strategies in MF-TDMA Satellite Systems*, Acta Aeronautica Et Astronautica Sinica, vol. 30 no. 9, pp. 1718-1726, September 2009.

[88] *psutil*, [online]. Available: https://pypi.org/project/psutil/

[89] Jose Mendes, et al., *Cellular Access Multi-Tenancy through Small Cell Virtualization and Common RF Front-End Sharing*, Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization, October 2017.

[90] Josef Weidendorfer, Markus Kowarschik, and Carsten Trinitis, *A Tool Suite for Simulation Based Analysis of Memory Access Behavior*, Proceedings of the 4th International Conference on Computational Science (ICCS), Krakow, Poland, June 2004.

[91] Julian Seward, and Nicholas Nethercote, *Using Valgrind to detect undefined value errors with bit-precision*, Proceedings of the USENIX'05 Annual Technical Conference, Anaheim, California, USA, April 2005.

[92] Intel Corporation, *Intel's Intrinsics Guide*, [online]. Available: https://software.intel.com/sites/landingpage/IntrinsicsGuide/

[93] J. Neyman and E. S. Pearson, *On the problem of the most efficient tests of statistical hypotheses*, Philosophical Transactions of the Royal Society of London, Series A. 231: 289-337, 1933.

[94] Janne J. Lehtomaki, Markku Juntti and Harri Saarnisaari, *CFAR Strategies for Channelized Radiometer*, IEEE Signal Processing Letters, vol. 12, no. 1, January, 2005.

[95] J. G. Proakis, *Digital Communications*, 5th Edition, McGraw-Hill, New York, 2001.

[96] Papoulis, Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th Edition, McGraw-Hill Europe, January 2002.

[97] M. H. DeGroot, *Probability and Statistics*, 2nd Edition, Addison-Wesley, 1986.

5

SCATTER PHY: An Open Source Physical Layer for the DARPA Spectrum Collaboration Challenge

In this chapter, we present an open source physical layer, which was employed in the DARPA Spectrum Challenge. This physical layer is built on top of the framework presented in the previous chapter and presents several improvements on it.

⋆ ⋆ ⋆

This chapter is based on the homonymous article by

Felipe A. P. de Figueiredo, Dragoslav Stojadinovic, Prasanthi Maddala, Ruben Mennes, Irfan Jabandzic, Xianjun Jiao, and Ingrid Moerman.

Published in to MDPI Electronics, vol. 8, no. 11, Nov. 2019.

Abstract DARPA, the Defense Advanced Research Projects Agency from the United States, has started the Spectrum Collaboration Challenge with the aim to encourage research and development of coexistence and collaboration techniques of heterogeneous networks in the same wireless spectrum bands. Team SCATTER has been participating in the challenge since its beginning, back in 2016. SCATTER's open-source SW defined physical layer (SCATTER PHY) has been developed as a standalone application, with the ability to communicate with higher layers through a set of well defined messages (created with Google's Protocol buffers)

and that exchanged over a ZeroMQ bus. This approach allows upper layers to access it remotely or locally and change all parameters in real time through the control messages. SCATTER PHY runs on top of USRPs based SW defined radio devices (i.e., devices from Ettus or National Instruments) to send and receive wireless signals. It is a highly optimized and real-time configurable SDR based PHY layer that can be used for the research and development of novel intelligent spectrum sharing schemes and algorithms. The main objective of making SCATTER PHY available to the research and development community is to provide a solution that can be used out of the box to devise disruptive algorithms and techniques to optimize the sub-optimal use of the radio spectrum that exists today. This way, researchers and developers can mainly focus their attention on the development of smarter (i.e., intelligent algorithms and techniques) spectrum sharing approaches. Therefore, in this chapter, we describe the design and main features of SCATTER PHY and showcase several experiments performed to assess the effectiveness and performance of the proposed PHY layer.

5.1 Introduction

In the era of a growing number of wireless communication networks and protocols, spectrum scarcity represents an ever-increasing challenge in the research community. The Defense Advanced Research Projects Agency (DARPA) has established the SC2 as a collaborative machine learning competition, where different teams from the entire world build their own wireless communication networks to compete with other teams in achieving the highest possible throughput, not only for their own team, but also for other teams coexisting in the same spectrum bands [1]. The teams were motivated to set up their networks to be able to not only coexist with other networks, but also use collaboration to ensure that each network maximizes its throughput while facilitating spectrum usage and securing its availability when necessary for other networks and systems. The teams would devise new spectrum access strategies, which would allow radio networks to collaborate autonomously and determine dynamically how the RF spectrum is being used, which enables radios to avoid interference and find better transmission opportunities [2]. The SC2 was run on Colosseum, a testbed with a large number of nodes equipped with SDR devices, where teams can deploy their networks and use SDRs for their wireless communication. The testbed is able to emulate any desired geographical topology of nodes with diverse physical environments, enabling DARPA to test teams' networks and their performance with and against other networks in a wide variety of scenarios [1].

The SCATTER team has designed and built a multi-layered system, with physical, MAC, AI, and other layers, which were developed and implemented as completely autonomous systems, but interconnected through a publish-subscribe messaging system known as ZeroMQ [3, 4]. ZeroMQ (also known as 0MQ or ZMQ) is a high-performance asynchronous messaging library, aimed for use in concurrent or distributed applications [5]. Therefore, SCATTER's open-source SW defined

PHY layer was designed as an independent system, able to transmit and receive packets over the air using USRPs SDR devices. The design of SCATTER PHY uses srsLTE, an existing SW implementation of the LTE protocols, as the basis for its communication and builds on top of it with various modifications to introduce features and mechanisms targeted for the SC2. The ultimate goal of these features is to present an API of the required physical layer functionality to other layers (mainly the MAC and AI layer) which are oblivious to the implementation details of those functions. For example, SCATTER PHY uses custom built FPGA filters to reduce OOBE and maximize channel utilization. When the MAC layer requests a change of the channel bandwidth, these filters are automatically adapted to accommodate the new bandwidth. Similarly, the physical layer provides spectrum measurement data averaged over time to the AI layer, which uses them to determine the most efficient channel selection algorithms. Spectrum measurement data are obtained by again using the custom built FPGA blocks, which perform FFT and average the obtained power spectral density over multiple measurements to reduce the amount of data sent to AI.

AI is of utmost importance to the SCATTER system, specially in the MAC design for dynamic spectrum access. During MAC scheduling, AI is used to predict spectrum usage and provide MAC with information to better schedule radio resources (channel/time-slot) and avoid collisions. During link adaptation, AI is responsible for optimizing Tx gain (power) and MCS. For technology recognition tasks, AI identifies and discriminates between competing teams in order to find the better strategy and maximize our score. MAC and AI modules were implemented by other students and are described in details in [3, 6–9]. The implementation and interactions between the AI and MAC modules are out of the scope of this chapter.

In this work, we describe the design and features of SCATTER PHY and experimentally verify the benefits achieved with the outlined design decisions. The SCATTER PHY layer, presented in this work, is an enhancement/extension of the framework presented in [10]. Different from [10], SCATTER PHY presents several enhancements on top of the existing framework.

Next, we list and describe the main contributions and differences of this work when compared to [10].

- Introduction of an open-source, highly optimized, and real-time configurable SDR based physical layer for the research and development of novel intelligent spectrum sharing schemes and algorithms. The main idea behind the development of the proposed physical layer is to provide developers and researchers with a starting point so that they can mainly focus their attention on developing novel intelligence and reasoning algorithms to combat spectrum scarcity. The presented physical layer can be employed in prototyping and experimentally testing intelligent algorithms for dynamic optimization of spectrum usage. Its source code can be found on its GitHub project page [11].

- Two concurrent bursty f-OFDM based PHY layers that operate independently of each other in full or half-duplex modes.

- A novel frame structure with a focus on higher throughput (with 32 MCS values, achieving more than 84 Mega bits per second (Mbps) for a configured PHY BW of 9 MHz and MCS 31) and with control signals being sent by the robust and more reliable M-sequences instead of LTE-like control channels.

- FPGA based FIR filters that improve/speed up the PHY processing performance while mitigating OOBE.

- Proposing a two-stage synchronization sequence detection algorithm, which uses a CA-CFAR algorithm in the second stage in order to improve the detection performance when compared to an approach that only employs plain correlation to detect the presence of the synchronization sequence. Our results show that the proposed two-stage approach improves subframe detection and synchronization in low SNR scenarios.

- CP based CFO estimation implemented with SIMD instructions and a more accurate and fine-grained complex exponential correction signal, which generates correction signals with frequencies closer to the estimated ones.

- Simple receive (Rx) combining scheme where the detected and synchronized signals coming from the two independent PHYs are combined for improved performance in low SNR scenarios.

- The effectiveness of the presented physical layer is verified through a comprehensive set of experiments and their respective results. We present and discuss the results of experiments including measurements of CPU and memory utilization, throughput, the effectiveness of a filtered form of the OFDM waveform, etc.

The remainder of the chapter is organized as follows: Section 5.2 presents some related pieces of work that apply artificial intelligence algorithms and methods to the spectrum scarcity/sharing problem. Section 5.3 describes the design of SCATTER PHY in detail and specifies its main features. Section 5.4 showcases the experimental verification of the achieved benefits. Section 5.5 concludes the chapter.

5.2 Related Work

In [12], the authors proposed the use of deep reinforcement learning to design a universal MAC protocol referred there as Deep-reinforcement Learning Multiple Access (DLMA). They showed through simulations that the proposed DLMA scheme was able to learn the optimal MAC strategy for harmonious co-existence with TDMA and ALOHA networks.

The authors of [13] employed a Deep Learning Neural Network (DLNN) for the slot prediction task. They demonstrated through simulations that their trained

DLNN did online learning and could accurately predict, through spectrum monitoring, the behavior of spectrum usage (i.e., slot transmissions) one second in advance in the context of MF-TDMA networks.

In the work presented in [14], the authors dealt with the problem involving the recognition of different RATs employing Deep Autoencoders (DAEs). Their DAEs model was used in a Semi-Supervised Learning (SSL) approach for the recognition of wireless technologies using raw IQ samples. The simulation results showed that the proposed DAEs approach achieved similar accuracy when compared with Supervised Learning (SL). They also demonstrated that their approach outperformed the SL approach for negative SNR values.

The authors of [15] modeled a collaborative power control scheme in an interference limited wireless network, where nodes shared the same spectrum band. They employed game theory in order to determine the optimal power allocation for every node. The simulation results showed that by collaborating (i.e., exchange of node information) with other nodes, it was possible to maximize not only a single node's performance, but also other nodes' performances.

In [16], the authors proposed the use of a Q-learning based dynamic optimal band and channel selection scheme, which took into account the surrounding wireless environments and system demands in order to maximize the available transmission time and capacity at the given time and geographic area. The simulation results showed that their solution dynamically chose a band and channel suitable for the required data rate and operated properly according to the desired system performance.

All the above described previous works heavily relied on simulations to access the performance of several artificial intelligence algorithms applied to the spectrum sharing problem. However, they lacked experimental validation of their approaches through the use of test-beds or real-world channels. For instance, after the DARPA challenge is over, the Colosseum test-bed, which was developed for the competition, will become a National Wireless Research asset and made available to the public for any kind of wireless experiments. With that in mind, since the inception of our PHY layer, we developed SCATTER PHY to be highly configurable and decoupled from the upper layers, i.e., the layers connected to our PHY layer can access it remotely and change a great number of parameters in real time through well defined control messages. Therefore, we hope, by making the SCATTER PHY layer available as an open-source project, to close the gap between simulation-only validations and real-world experiments.

5.3 SCATTER PHY

Figure 5.1 depicts the architecture of SCATTER PHY at a high level. The figure shows the several HW and SW layers making up SCATTER PHY and the different threads running inside of them.

Black arrows indicate the exchange of control and/or information between the threads, while red dashed arrows indicate data paths. SCATTER PHY is a bursty

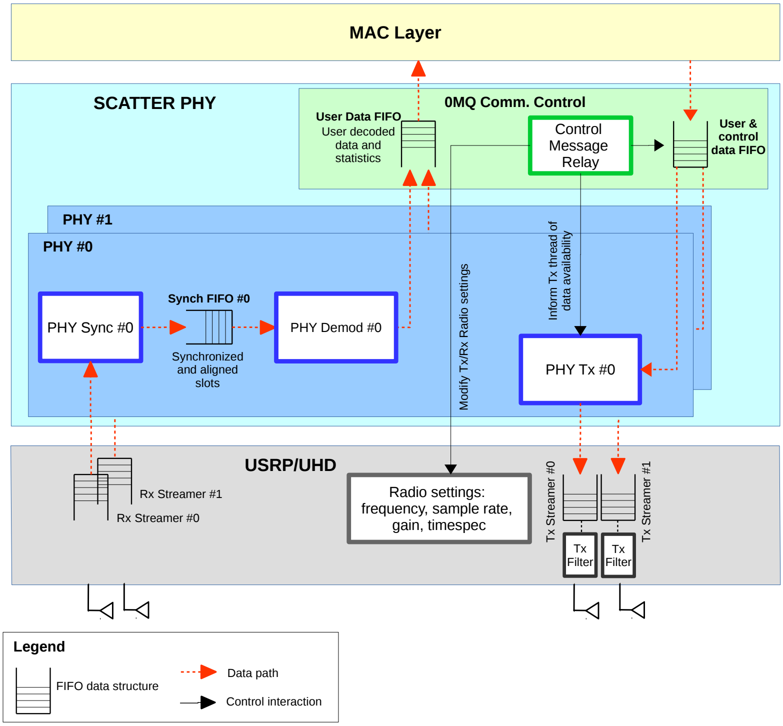


Figure 5.1: High level threading architecture of SCATTER PHY.

based PHY that transmits data bursts in short intervals. Each bursty transmission interval contains at least one or several transport units, which we call subframes. A subframe is a container through which control and user data are exchanged, over the air, in the network. Therefore, a subframe is the basic transmission unit of SCATTER PHY, and each one is 1 ms long.

The SCATTER PHY is an open-source SW defined physical layer [11]. Its implementation makes use of the UHD SW API [10, 17], which makes it possible for the PHY to run on top of several Ettus SDR devices such as the Ettus USRPs X family of SDRs and NI's RIO devices [18, 19]. SCATTER PHY accesses the SDR device through the UHD driver and its APIs [20]. As can be seen in the figure, the independent PHY layers are connected to a module responsible for managing the exchange of data, control, and statistics messages with the MAC layer, and the module is known as the ZeroMQ Data/Control Communications module. The exchange messages with the MAC layer are performed through a ZeroMQ bus, also known as 0MQ [5].

In order to establish communications independently with the MAC layer, i.e., decouple the implementation of both layers, a well defined interface (i.e., a set of pre-defined messages) was designed with Google's Protocol Buffers (protobuf) [21]. Protobuf was used for data serialization and worked perfectly with the

0MQ messaging library [5]. Together, protobuf and 0MQ created a means for a distributed and decoupled exchange of control, statistics, and data messages between SCATTER PHY and MAC layers. SCATTER PHY provides the MAC layer with the ability of real-time configuration of all its parameters and reading of several statistics it measures through the implementation of the 0MQ push-pull message-exchanging pattern [5]. The implementation of this message-exchanging pattern provides upper layers with the ability of locally to configure/read PHY parameters/statistics remotely, i.e., upper layers (e.g., MAC layer) do not need to be run on the same physical radio. The 0MQ push-pull pattern allows MAC and PHY to implement a non-blocking message communications paradigm, i.e., MAC and PHY can exchange data and control messages without having their processing blocked for long periods, and with this paradigm, each layer decides when to check the availability of a new message. The implementation of the 0MQ push-pull pattern and the adoption of the protobuf message library allow SCATTER PHY to be designed and implemented totally independently and decoupled from the MAC layer, i.e., SCATTER PHY's implementation decisions do not impose any kind of constraints on the programming language, SW, or HW employed by the MAC layer.

SCATTER PHY presents the following set of main features:

- Bursty transmissions: Short periods of activity (i.e., discontinuous transmissions) allow a better/improved usage of the available spectrum BWs. It makes it possible for the radios/networks to coordinate the spectrum access/usage among them in a collaborative, intelligent, or even opportunistic manner.

- Dual-concurrent PHYs: A Multi-Concurrent-Frequency Time-Division Multiple Access (McF-TDMA) scheme can be implemented by the MAC layer by having two PHYs simultaneously transmitting and receiving at independent frequencies. This ability for concurrent allocations allows for a smarter spectrum utilization as vacant disjoint frequency chunks can be concurrently used.

- FPGA based filtered transmissions: Filtering the transmitted signal effectively minimizes OOBE, which allows for a better spectrum utilization, once radios can have their transmissions nearer to each other in the frequency domain, helping to mitigate the wastage of spectrum bands.

- Out-of-band full-duplex operation: Both PHYs operate totally independently of each other, meaning that Tx and Rx modules are able to transmit and receive at different channels, set different Tx and Rx gains, and use different PHY BWs.

- Timed-commands: This feature allows the configuration in advance of the exact time to (i) start a transmission and (ii) change Tx/Rx frequencies/gains. This allows the MAC layer to implement a TDMA scheme.

- Improved synchronization: Through the design and implementation of a novel two-stage synchronization algorithm, it is possible to improve the packet detection ratio when compared to the single-stage approach.
- Fine-grained CP based CFO estimation: By generating a complex exponential sequence with smaller sampling steps, it is possible to synthesize the frequency offset more accurately, consequently improving the correction.
- Channel emulator: Through the implementation of an abstraction layer between the USRPs HW and the SW based PHY, it is possible to create a channel emulator that connects the transmission side to the receiving side without the necessity of any HW. Through the use of the channel emulator, it is possible to test novel PHY layer algorithms with the addition of white Gaussian noise, multi-paths, frequency offsets, etc.

SCATTER PHY was implemented based on the srsLTE library [22]. srsLTE is an open-source and free LTE SW based library that was developed by the SRS Limited company [22]. It implements LTE release 10 and a few NB-IoT and 5G features. As SCATTER PHY was built on an LTE-like library, it consequently absorbed and evolved on top of the existing 4/5G features.

We employed OFDM as the SCATTER PHY waveform. OFDM is a full blown technology that is present in a huge number of commercial products due to various advantages it offers such as low implementation complexity, robustness to severe multi-path fading, simple channel estimation, easy integration with MIMO technology, etc. [23]. User and control data are mapped into subcarriers over 14 OFDM symbols spanning 1 ms. The control data signal carries the used MCS for the current transmission and the number of subsequent subframes with user data modulated with that specific MCS.

The control data signal is used by the independent PHY receivers to detect the number of allocated RBs automatically (i.e., control is embedded into the transmitted data signal), the MCS value selected to transmit data of a specific device/radio, and the location of the allocated RBs in the subframe's resource grid.

By embedding the control information into the transmitted signal, the MAC layer does not need to know in advance the number of subframes and MCS in a given COT. After successful user data decoding, each one of the PHYs independently notifies the MAC layer about the corresponding decoded MCS value, the number of received data bytes, and decoding statistics (e.g., RSSI, SINR, number of detection/decoding errors, etc.).

SCATTER PHY works with two types of subframes, namely synchronization and data-only subframes. A synchronization subframe carries the synchronization signal, reference signal, control, and user data. A data-only subframe carries the reference signal and user data. The synchronization signal is a 72 symbol long sequence that is generated using Zadoff-Chu (ZC) sequences [24, 25]. Therefore, the synchronization sequence is generated according to:

$$x_s = e^{-j\frac{\pi u n(n+1)}{N_{\text{Synch}}}}, 0 \leq n \leq N_{\text{Synch}} - 1, \tag{5.1}$$

where n is the sequence sample index, u is the ZC sequence index, and N_{Synch} is the length of the synchronization sequence.

The control signal is based on M-sequences [26], where two M-sequences, each of length 31, are used to transmit the information necessary to decode the user data. The reference signal is used to estimate the channel and then equalize the received signal so that channel interference to the desired signal is minimized.

The SCATTER PHY allows bursty transmissions with a variable COT, i.e., the number of subframes to be transmitted in a row without any idle time (i.e., gap) among consecutive subframes is variable and depends on how many bytes the MAC layer has to transmit. The number of subframes in a COT is calculated based on the number of data bits to be transmitted, PHY BW, and MCS parameters forwarded by the MAC layer to each PHY layer in the control messages. The smallest COT value is equal to 1 ms and corresponds to the transmission of a synchronization subframe. Having a variable COT allows SCATTER PHY to support various types of traffic loads. Every subframe carries a pre-defined number of bytes. The number of bytes is calculated based on the subframe type, selected MCS, and configured PHY BW.

As can be seen in Figure 5.1, each one of the PHY modules is divided into three sub-modules, namely PHY Rx synchronization, PHY Rx demodulation, and PHY Tx. Each one of the sub-modules runs on a standalone and exclusive thread. A multi-threaded PHY implementation allows independent, time-consuming, and critical processing tasks to be executed concurrently (i.e., simultaneously), improving efficiency and computing performance. In combination with multi-core enabled CPUs, the multi-threaded PHY modules straightforwardly support communications in full-duplex mode, i.e., each individual and independent PHY is able to receive and transmit simultaneously at different frequencies. The direct consequence of the full-duplex mode is that it results in higher throughput.

Before proceeding with the description of SCATTER PHY, it is vital to notice that SCATTER PHY development was guided by DARPA's provided test environment, i.e., their testbed, called Colosseum. Colosseum's channel model supported the addition of noise and attenuation of the transmitted signals and did not support MIMO, more specifically spatial multiplexing (its channel matrix had a rank equal to 1). We decided on the dual PHY approach because it didn't support MIMO and wanted to maximize the system's throughput. Dual PHY gives flexibility and allows the system to reach higher throughput values than the PHY version presented earlier. The MAC layer used in our system was designed to support such features, i.e., full-duplex [6]. It is also important to highlight that Colosseum had problems with the calibration of the radios and the channel emulator making it up, so it suffered a lot with CFO. Therefore, that is why we have also proposed improvements on CFO estimation and correction in this work.

Each PHY Tx thread (#0 and #1) is responsible for modulating and transmitting data (i.e., control and user data). On the other hand, each PHY Rx synchronization thread is responsible for detecting the Synchronization (Synch) sequence, decoding the control data, frequency offset estimation/correction, and time-alignment of the detected subframes. Detection of the Synch signal is carried out through a two-stage detection algorithm, which at the first stage correlates the received signal with a locally stored version of the synchronization subframe with no data and control signals. If the Peak-to-Side-lobe Ratio (PSR) is greater than a constant threshold, then the second stage applies a CA-CFAR algorithm to the OFDM symbol carrying the Synch signal [24]. The two-stage approach employed by SCATTER PHY improves the Synch signal detection when compared to a detection approach that only uses the PSR of the correlation calculated at the first stage. The two-stage approach employed by SCATTER PHY is described in Appendix C.1.

The CFO estimation task is split into coarse and fine estimations/corrections, where the coarse estimation is based on the Synch signal and the fine estimation is based on the CP portion of the OFDM symbols [27]. The integer part of the frequency offset (i.e., integer multiples of the subcarrier spacing) is estimated and corrected by the coarse CFO algorithm, which is based on the maximization of the correlation of the received synchronization signal with several locally generated frequency offset versions of it. On the other hand, the fractional frequency offset (i.e., offset values less than one half of the subcarrier spacing) is estimated and corrected by the fine CFO algorithm, which is based on the phase difference of the correlation between the CP and the last part of the OFDM symbol (i.e., the portion used to create the CP). Integer and fractional CFO estimation methods are described in Appendix D.1.

Each PHY Rx demodulation thread is responsible for OFDM demodulation (CP removal and FFT processing), i.e., user data demodulation, channel estimation/equalization, resource de-mapping, symbol demodulation, de-scrambling, de-interleaving/de-rate matching, turbo decoding, de-segmentation, and finally, CRC checking. Each one of the independent PHYs receives control and data messages from the 0MQ Communications Data/Control module. Decoded user data and Tx/Rx statistics are sent directly to the MAC layer through the 0MQ bus within a protobuf pre-defined message. Figure 5.2 presents the block-diagram of SCATTER PHY. It shows the blocks making up the receiver (PHY Synch and PHY Demodulation (Demod) threads) and transmitter (PHY Tx thread) sides of each independent PHY.

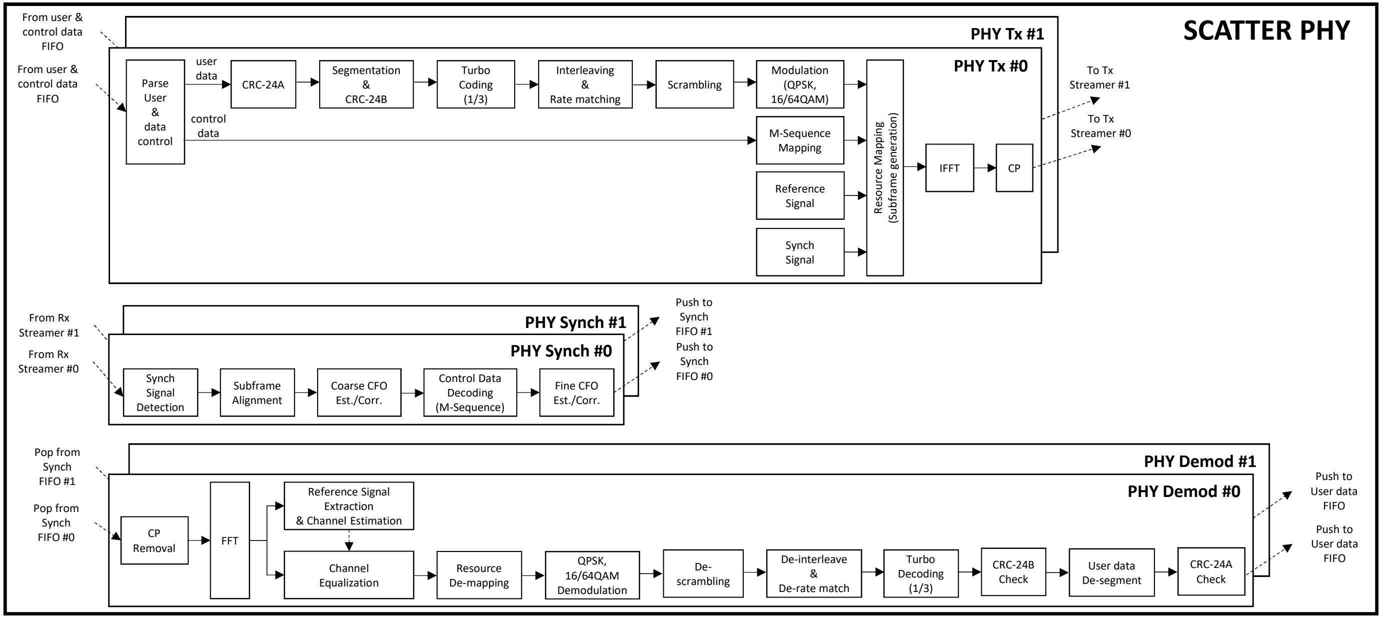


Figure 5.2: Block-diagram showing the modules making up the transmitter and receiver sides of the SCATTER PHY.

Table 5.1: OFDM modulation parameters.

PHY BW (MHz)	1.26	2.7	4.5	9
Subframe duration (ms)	1			
subcarrier spacing (kHz)	15			
Sample rate (Msps)	1.92	3.84	5.76	11.52
FFT size	128	256	384	768
Useful subcarriers	84	180	300	600
OFDM symbols per subframe	14			
CP duration (μs)	4.69 (12 symbols)/5.21 (2 symbols)			

When it comes to numerology, SCATTER PHY adopts a subcarrier spacing of 15 kHz, a CP of 5.2 μs, and supports 1.26, 2.7, 4.5, and 9 MHz BWs, which are equivalent to the LTE BWs of 6, 15, 25, and 50 resource blocks, respectively [23]. Each one of the PHY's transmission channel BW can be configured in real time, through the Tx control messages or via command line at the system's startup. The OFDM modulation parameters adopted by SCATTER PHY are summarized in Table 5.1.

The only difference between the OFDM modulation parameters listed in Table 5.1 and the LTE ones [23] is the FFT size and, consequently, the resulting sampling rate for the PHY BWs of 4.5 and 9 MHz. The reason for employing these non-standard sampling rates is to reduce the computational load of SCATTER PHY and to reduce the data transfer rate required to move base-band IQ (*i.e.*, In-phase/Quadrature) samples between the RF front-end (*i.e.*, the USRPs device) and the base-band processor (i.e., the host PC) and vice versa. For example, for a PHY BW of 4.5 MHz, the standard LTE sampling rate would be equal to 512 FFT points $\times$ a subcarrier spacing of 15 kHz $=$ 7.68 MHz. However, out of the 512 subcarriers, only 300 are useful subcarriers, where the remaining ones are left unused as guard-band subcarriers. Therefore, by employing a smaller FFT size and, consequently, a lower sampling rate, while keeping the same subcarrier spacing, SCATTER PHY can still successfully demodulate all occupied subcarriers. Therefore, the computational load is lower due to the fact that fewer base-band samples have to be moved between the USRPs and the host PC.

The frame structure employed by SCATTER PHY is presented in Figure 5.3. As can be seen, signals regarding synchronization and control data are only added to the first subframe of a COT, while the subsequent subframes carry only reference signals and data symbols. With this frame structure, synchronization (i.e., detection of synchronization signal, time-alignment, and CFO estimation/correction) and control data decoding only happen for Subframe $\#0$, i.e., the synchronization subframe. The differences between the proposed frame structure and LTE's are (i) no modulated control data (i.e., LTE's PDCCH channel) being added to first OFDM symbols, (ii) a longer synchronization signal, which improves the subframe detection performance, (iii) control data sent via M-sequences, which makes the control data decoding more robust against interference and noise, and (iv) no

Table 5.2: MCS and their respective coding rates.

		Code Rate			
MCS	**Modulation**	**1.26 MHz**	**2.7 MHz**	**4.5 MHz**	**9 MHz**
0	QPSK	0.0857	0.0900	0.0940	0.0940
1	QPSK	0.1143	0.1200	0.1220	0.1220
2	QPSK	0.1357	0.1467	0.1480	0.1480
3	QPSK	0.1714	0.1933	0.1920	0.1920
4	QPSK	0.2107	0.2400	0.2400	0.2440
5	QPSK	0.2571	0.2933	0.2960	0.2920
6	QPSK	0.3071	0.3400	0.3440	0.3440
7	QPSK	0.3571	0.4000	0.4160	0.4130
8	QPSK	0.4071	0.4667	0.4640	0.4690
9	QPSK	0.4714	0.5200	0.5360	0.5330
10	16QAM	0.2357	0.2600	0.2680	0.2665
11	16QAM	0.2571	0.2933	0.2920	0.2905
12	16QAM	0.3000	0.3267	0.3280	0.3305
13	16QAM	0.3357	0.3667	0.3800	0.3825
14	16QAM	0.3857	0.4267	0.4290	0.4298
15	16QAM	0.4286	0.4667	0.4770	0.4718
16	16QAM	0.4429	0.5000	0.5090	0.5078
17	64QAM	0.2952	0.3333	0.3393	0.3385
18	64QAM	0.3238	0.3600	0.3553	0.3625
19	64QAM	0.3524	0.3911	0.4033	0.4087
20	64QAM	0.3905	0.4411	0.4353	0.4407
21	64QAM	0.4286	0.4678	0.4727	0.4727
22	64QAM	0.4571	0.5122	0.5047	0.5100
23	64QAM	0.4952	0.5478	0.5570	0.5642
24	64QAM	0.5333	0.5833	0.5970	0.6042
25	64QAM	0.5714	0.6189	0.6210	0.6308
26	64QAM	0.5905	0.6633	0.6690	0.6770
27	64QAM	0.6190	0.6900	0.7010	0.7010
28	64QAM	0.6571	0.8056	0.8067	0.8178
29	64QAM	0.6952	0.8322	0.8280	0.8552
30	64QAM	0.7333	0.8617	0.8493	0.8925
31	64QAM	0.7714	0.8883	0.8707	0.9240

division of the subcarriers into resource blocks, once all subcarriers are allocated to a single user only. As shown in Figure 5.3, and differently from LTE, SCATTER PHY's frame structure does not carry control data in the first OFDM symbols, which, consequently, gives more room, i.e., subcarriers, for user data transmission. Therefore, it is possible to achieve higher throughput when compared to the LTE standard.

Table 5.3: List of real-time configurable parameters and statistics provided by SCATTER PHY.

Message	Parameter	Type	Unit	Range
Tx control	PHY Identity (ID)	uint32	-	0-1
	MCS	uint8	-	0-31
	Tx gain	uint32	[dB]	depends on HW [1]
	Tx channel	uint32	-	≥ 0
	Tx PHY BW	uint8	MHz	0-3 [2]
	Transmission timestamp	uint64	s	≥ 0
	User Data length	uint32	-	> 0
	User data	uchar[]	-	uchar range
Rx control	PHY ID	uint32	-	0-1
	Rx channel	uint32	-	≥ 0
	Rx gain	uint32	[dB]	depends on HW [1]
	Rx PHY BW	uint8	MHz	0-3 [2]
	Max. # of turbo decoder iterations	uint32	-	uint32 range
	Enable Rx combining	boolean	-	True/False
Rx statistics	PHY ID	uint32	-	0-1
	CQI	uint8	-	0-15
	RSSI	float	dBW	float range
	Noise	float	dBW	float range
	Decoded MCS	uint8	-	0-31
	Subframe error counter	uint32	-	≥ 0
	Decoding time	uint32	ms	≥ 0
	Data length	uint32	-	≥ 0
	# of turbo decoder iterations	uint32	-	uint32 range
	Received data	uchar[]	-	uchar range
Tx statistics	PHY ID	uint32	-	0-1
	Coding time	uint32	ms	≥ 0
	Number of transmitted subframes	uint32	-	≥ 0

[1] This parameter depends on the installed USRPs daughter board model [20]. [2] The numbers correspond to the following BWs: 1.26, 2.7, 4.5, and 9 MHz, respectively.

The PDU employed by SCATTER PHY is known as the TB. A TB is the container carrying data that are created at the MAC layer and transferred to the individual PHYs to be encoded and transmitted. One TB is composed of a number of bytes that can be accommodated inside a 1 ms long subframe, given the configured MCS and PHY BW parameters. Therefore, given the PHY BW and the desired MCS, the MAC layer can calculate the number of bytes that can be handled by a 1 ms long subframe. Table 5.2 presents the coding rate for each one of the 32 defined MCS values.

The differences between SCATTER PHY's modulation code scheme and LTE's are (i) 32 different MCS values instead of the 29 defined in the LTE standard, where the three additional MCS values allow SCATTER PHY to reach higher data rates in high SNR scenarios, and (ii) the greater number of subcarriers per subframe, which are used for user data transmission and, consequently, increase the final achieved throughput. However, even though being able to carry more user

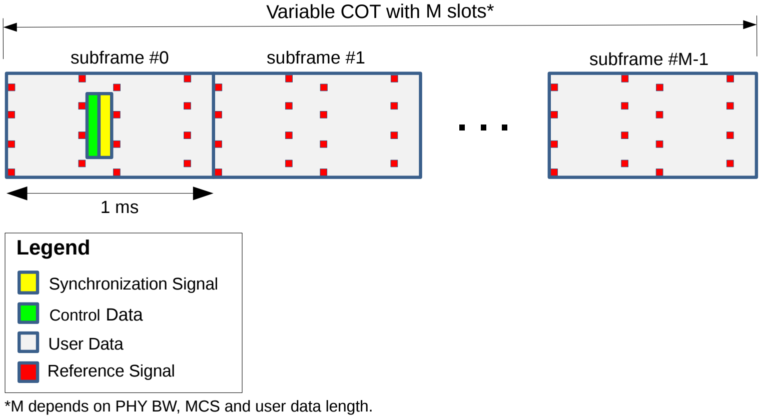


Figure 5.3: SCATTER PHY frame structure with variable COT.

data per subframe, we kept the same coding rate of the LTE standard for MCS values ranging from zero to 28.

Communications between SCATTER PHY and the MAC layer are performed through four pre-defined (i.e., protobuf) messages. The first two messages, named Rx and Tx control, are employed, as the name suggests, to control/configure subframe reception and transmission, respectively. The parameters transported by these two control messages are set and sent to the individual PHYs by the MAC layer before the transmission of every new COT, hence allowing on-line parameter configuration. The other two messages, named Rx and Tx statistics messages, are employed to provide the MAC layer with real-time feedback from each independent PHY, providing critical information that is necessary for such a layer to take important actions.

Tx control messages transport the TBs (i.e., user data) to be transmitted and Tx parameters regarding that transmission, that is PHY ID, Tx gain, Tx channel, data length, MCS, Tx PHY BW, and transmission timestamp. The transmission timestamp parameter allows time scheduled transmissions, which consequently enable the MAC layer to implement a MF-TDMA medium access scheme. Rx control messages are employed to configure Rx gain, the Rx channel, the maximum number of turbo decoder iterations, enable or disable Rx combining, and Rx PHY BW of a specific PHY, addressed through the PHY ID parameter. The PHY ID parameter is used to specify for each one of the two PHYs a control message.

Tx statistics messages inform the MAC layer about transmission statistics like data coding-time and the total number of transmitted subframes of each independent PHY. Rx statistics messages transport the received data, PHY ID, and reception statistics regarding the received data such as RSSI, CQI, decoded MCS, decoding time, subframe error counter, number of turbo decoder iterations, etc. The real-time configurable parameters and statistics offered by SCATTER PHY are summarized in Table 5.3.

5.3.1 COT-Based Filtering

As mentioned in the previous chapter, OFDM based wave-forms are not suited for spectral coexistence [28] due to their poor spectral localization.

Therefore, in order to guarantee lower OOBE, i.e., improved spectral-localization, and still maintain the orthogonality of the OFDM symbols, it is necessary to filter the generated subframes before their transmission [10]. The filtering process is applied to each COT of each PHY independently. The subframes comprising a COT are generated at the SW level and then filtered at the HW level, by a 128 order FPGA based FIR filter. The COT based filtering improves the closer coexistence with other radios (either belonging to our team or others), allowing radio transmissions to be closer in frequency. The filter used in SCATTER PHY was designed and explained in details in [10].

This co-design SW/HW is used so that fast processing high order filters can be implemented adding up very low latency to the transmission chain and still allowing the flexibility of the SW defined PHYs.

The coefficients of the filters applied to the COT are automatically selected in real time based on the Tx PHY BW parameter carried by the Tx control message. The coefficients are automatically chosen as the filters' cut-off frequencies have to match and exactly filter the desired signal's bandwidth, i.e., the configured PHY transmit BW.

The normalized FIR filter's coefficients used in the COT based filtering are given in time domain by (4.1), where the signals $p(n)$ and $w(n)$ are given as

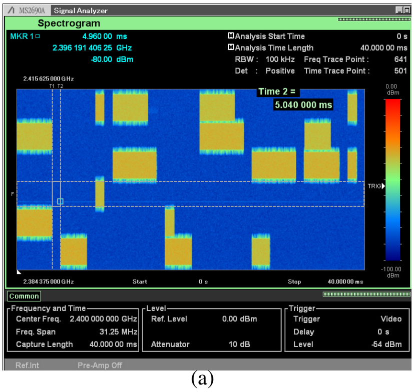
$$p(n) = \begin{cases} \sin\left(\frac{\pi[N_{\text{U}}+N_e]n}{N_{\text{FFT}}}\right) / \frac{\pi[N_{\text{U}}+N_e]n}{N_{\text{FFT}}}, & n \neq 0, \\ 1, & n = 0. \end{cases} \tag{5.2}$$

$$w(n) = \left\{\frac{1}{2}\left[1 + \cos\left(\frac{2\pi n}{L-1}\right)\right]\right\}^{0.6}, \tag{5.3}$$

where N_{U} is the number of useful subcarriers (see Table 5.1), N_{FFT} is the length of the FFT used in the OFDM modulation (see Table 5.1), N_e is the excess bandwidth in the number of subcarriers, L is the length of the FIR filter, and $-\frac{(L-1)}{2} \leq n \leq \frac{(L-1)}{2}$. The excess bandwidth is used to extend the flat region of the filter so that the subcarriers at the left and right borders of the OFDM symbols suffer less attenuation.

5.3.2 Benefits of SCATTER PHY

In this section, we describe the two main advantages of SCATTER PHY. The first advantage is that the COT based filtering mitigates OOBE, making SCATTER PHY more spectrally efficient. OOBE might interfere with radios using adjacent channels, decreasing the quality of the signals received by them. Therefore, the interference caused by OOBE impacts directly on the throughput achieved by those radios. The reduced OOBE allows SCATTER PHY to operate closer, in the frequency domain, to radios transmitting at adjacent channels. This, in turn, reduces



MKR 1 □ 4.960 00 ms
2.396 191 406 25 GHz
-80.00 dBm

Analysis Start Time 0 s
Analysis Time Length 40.000 00 ms
RBW : 100 kHz Freq Trace Point : 641
Det : Positive Time Trace Point : 501

Frequency and Time
Center Freq. 2.400 000 000 GHz
Freq. Span 31.25 MHz
Capture Length 40.000 00 ms

Level
Ref. Level 0.00 dBm
Attenuator 10 dB

Trigger
Trigger Video
Delay 0 s
Level -54 dBm

(a)

MKR 1 □ 3.920 00 ms
2.396 142 578 13 GHz
-80.85 dBm

Analysis Start Time 0 s
Analysis Time Length 40.000 00 ms
RBW : 100 kHz Freq Trace Point : 641
Det : Positive Time Trace Point : 501

Frequency and Time
Center Freq. 2.400 000 000 GHz
Freq. Span 31.25 MHz
Capture Length 40.000 00 ms

Level
Ref. Level 0.00 dBm
Attenuator 10 dB

Trigger
Trigger Video
Delay 0 s
Level -54 dBm

(b)

*Figure 5.4: Comparison of the McF-TDMA scheme with and without filtering. (**a**) FPGA based filters disabled. (**b**) The* 128 *order FPGA based filters enabled.*

spectrum band wastage by increasing the spectral efficiency, as depicted in Figure 4.9. The second advantage offered by SCATTER PHY is that it is possible to configure all PHY parameters in real time through the control messages. This makes SCATTER PHY a very flexible and agile SDR based PHY, allowing higher layers to configure the available parameters based on spectrum band availability, for instance.

5.4 Experiment Results

In this section, we demonstrate the effectiveness and usability of SCATTER PHY by presenting the results of several experiments. The experiments were executed with SCATTER PHY running on servers equipped with Intel Xeon E5-2650 v4 CPUs (@2.2 GHz, 30 M cache, 9.60 GT/s QPI, turbo, HT, 12 cores/ 24 threads, 105 Watts), 128 GB of RAM memory connected to x310 USRPs with 10 Gigabit Ethernet links, and equipped with CBX-120 RF daughter-boards [30].

Figure 5.4 depicts two spectrograms (i.e., the visual representation of the spectrum of frequencies of a signal as it varies over time) saved for 40 ms over a 31.25 MHz bandwidth with both PHYs set to operate concurrently, at each instant, at two of six 4.5 MHz channels. This experiment intends to show that SCATTER PHY is able to generate two concurrent and independent transmit channels per node. For this experiment, each PHY transmits at randomly selected channels a random number of subframes. The number of transmitted subframes, i.e., COT, and the channel number are randomly selected between the ranges of 1–3 and 0–5, respectively. Here, a gap of 1 ms between consecutive transmission is used. The figures were saved with the center frequency of both Tx PHYs set to 2.4 GHz, equal Tx gains of 3 dB, and both USRPs Tx outputs connected to the signal-analyzer via an RF combiner and a coaxial cable presenting a 20 dB attenuation. As can be

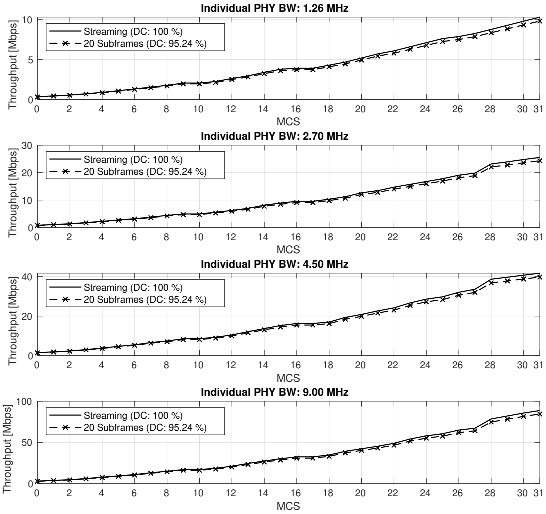


Figure 5.5: SCATTER PHY throughput for several Tx BW and MCS values.

seen in both figures, Figure 5.4(a) and Figure 5.4(b), SCATTER PHY was able to independently transmit at two distinct channels with a different number of subframes. Figure 5.4a shows the case where no filtering was enabled. As can be noticed, OOBE might cause interference to radios operating at adjacent channels and consequently decrease their throughput. Figure 5.4b shows the case when the 128 order FPGA-based FIR filters were enabled. As can be noticed, when filtering was enabled, OOBE was mitigated. Therefore, by enabling filtering, the interference that could be impacting radios operating at adjacent channels was mitigated. Moreover, another direct consequence of using filters was that the channel spacing, i.e., the guard-band between adjacent channels, could be decreased, decreasing the wastage of spectrum band, consequently. These results were saved with an Anritsu MS2690A Signal Analyzer. In the experiment, the signal analyzer sat beside the USRPs HW. In this experiment, the USRPs's Tx outputs were connected to the signal analyzer via coaxial-cables connected to a signal combiner and an attenuator with 20 dB of attenuation.

Figure 5.5 shows measurements of the throughput, which were taken with SCATTER PHY working in full-duplex mode, where the two independent PHYs simultaneously and concurrently receive and transmit, for several MCS and PHY BW values. The full-duplex mode was used so that we could check if it somehow impacted the throughput measurements, once, in this mode, SCATTER PHY was fully utilized. The throughput measurements were taken for COTs of 20 ms with gaps of 1 ms between subsequent COT transmissions, which resulted in a duty cycle of 95.24 %. The final throughput was calculated as the average over 10 throughput measurement intervals of 10 seconds each. For each measurement interval, the number of bits received during that interval was counted and then di-

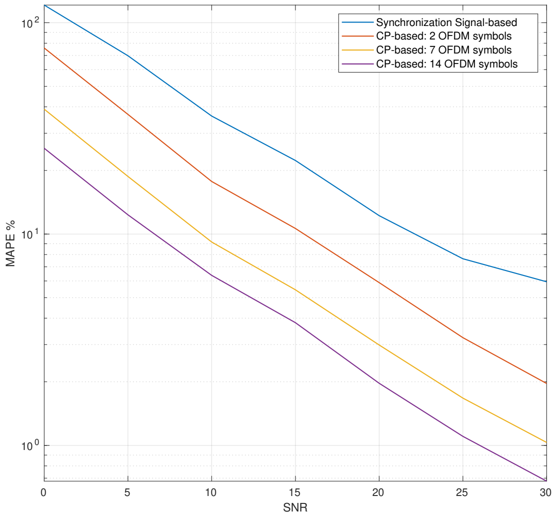


Figure 5.6: Comparison of CFO estimation error Mean Absolute Percentage Error (MAPE) in the AWGN channel.

vided by the length of the interval, resulting in a throughput measurement. This procedure was repeated 10 times, and then, the individual measurements were averaged. The SNR on the link was set to 30 dB so that the packet reception rate for all MCS values was equal to one. For the sake of comparison, we added to the figure the theoretical maximum throughput achieved when a duty cycle of 100 $\%$ was used; we called it streaming mode, due to the fact that there was no gap between subsequent subframes. The theoretical maximum throughput was obtained by dividing the size of the TB, given in number of bits for each MCS value, by the subframe time interval, i.e., 1 ms. As expected, the throughput measured with SCATTER PHY was close to that of the streaming mode for all PHY BW and MCS values, achieving more than 84 Mbps for MCS 31 and a PHY BW of 9 MHz. Moreover, full-duplex mode operation had no visible impact on SCATTER PHY's achieved throughput. This result comes from the fact that SCATTER PHY ran on a powerful server, with 12 cores and 128 GB of RAM memory.

Figure 5.6 compares the MAPE of the fine CP based fractional CFO estimation method against fractional CFO estimation based on the synchronization signal. The MAPE for the CFO estimation is defined as:

$$\text{MAPE } \% = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{\text{CFO}^{\text{est.}}(i) - \text{CFO}(i)}{\text{CFO}(i)} \right|, \tag{5.4}$$

where $\text{CFO}^{\text{est.}}(i)$ is the estimated CFO value for the i^{th} trial, $\text{CFO}(i)$ is the randomly generated CFO, which is applied to the transmitted signal, for the i^{th} trial, and N is the number of trials over which the CFO is averaged. The results were obtained by connecting the Tx port to the Rx port of the same USRPs so that the

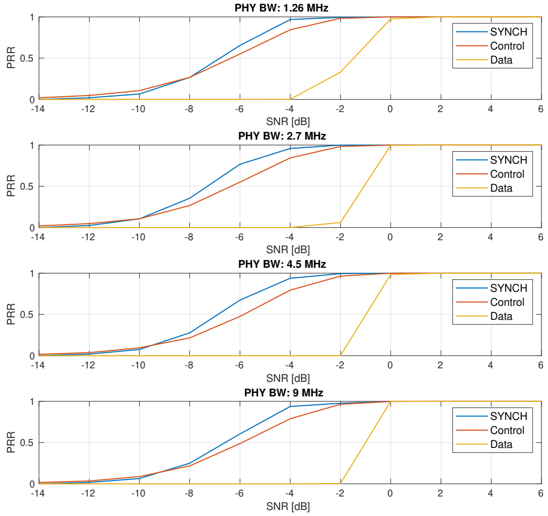


Figure 5.7: Comparison of the PRR for the different signals carried by a SCATTER PHY subframe.

frequency offset caused by the HW was minimal or nonexistent and adding AWGN plus the desired frequency offset at the SW level just before the subframes were transmitted. The CFO applied to the signal was drawn from a uniform distribution varying from -7.5 kHz to $+7.5$ kHz, i.e., $+/-$ half subcarrier spacing. As can be seen, the fine CFO estimation algorithm, which was based on the CP portion of the OFDM symbols, outperformed the Synch based CFO algorithm even when only two consecutive CPs were averaged. Additionally, we see that the performance of the CP based estimation improved as the number of averaged CPs increased; however, the downside of averaging more CPs was an increase in the processing time. The CP based CFO estimation method employed by SCATTER PHY was an improvement over the CFO estimation implemented by srsLTE. The SCATTER PHY implementation employed SIMD instructions in order to decrease the CFO estimation and correction computational time, allowing it to use more averaged CPs. Additionally, SCATTER PHY implemented a more accurate and fine-grained complex exponential signal for generating the correction signal, making it possible to generate signals with frequencies that were closer to the estimated ones.

Figure 5.7 presents the comparison of the PRR for each one of the signals carried by a SCATTER PHY subframe, namely synchronization, control, and data signals over several SNR and BW values. The SNR was calculated based on the power of a 1 ms subframe; therefore, before adding noise to the transmitted subframe, the subframe power was calculated, and then, the necessary noisy power to achieve the desired SNR was calculated. The MCS used for modulating the user data was set to zero, which was the most robust coding scheme, allowing SCATTER PHY to decode data in low SNR scenarios. The purpose of this ex-

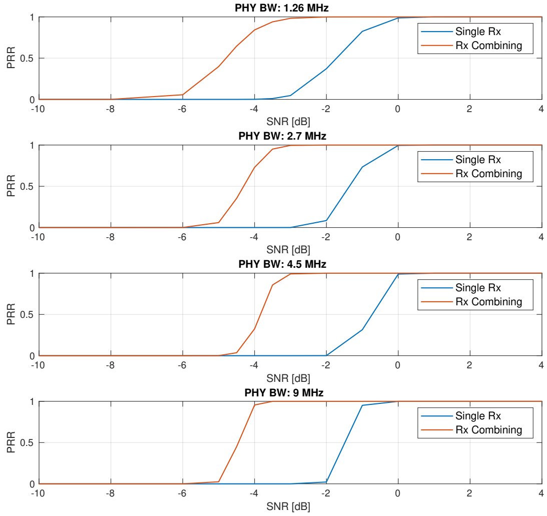


Figure 5.8: Simple Rx combined with SCATTER PHY for several PHY BW values.

periment was to identify the lowest possible SNR at which SCATTER PHY could still correctly decode the user data. This experiment was run by adding a channel emulator between the Tx and Rx sides of a single PHY instance. At the Tx side, the generated subframes, instead of being sent to the USRP HW, were sent to an abstraction layer that emulated the HW and added AWGN noise to the transmitted signal. Next, the abstraction layer transferred the noisy signal to the receiving side of the PHY. The PRR was averaged over 10^4 trials, where at each trial, the Tx side of the PHY sent a single synchronization subframe. As can be seen, synchronization and control signals had a better PRR performance than that of the data decoding; however, SCATTER PHY's PRR performance was limited by the ability to decode correctly the data section of a subframe. Additionally, we see that the data PRR was better for the 1.26 MHz case, which was due to the fact that compared to the other BW values, MCS 0 for the 1.26 MHz case carried more redundancy bits, as shown by Table 5.2, making it more robust against noise. The other two signals, synchronization and control, presented similar PRR curves for all BW values. Moreover, it is noticeable that the data PRR was equal to 1 for SNR values greater than or equal to 0 dB.

Figure 5.8 presents the result of the comparison of single Rx data decoding and that of a simple Rx combining scheme that can be implemented with SCATTER PHY. The figure shows data decoding PRR results for both receiving schemes. Here in this case, after synchronization (Synch signal detection, CFO estimation/-correction, control data decoding, and subframe alignment), the two independently synchronized subframes were combined through a simple average of both sub-

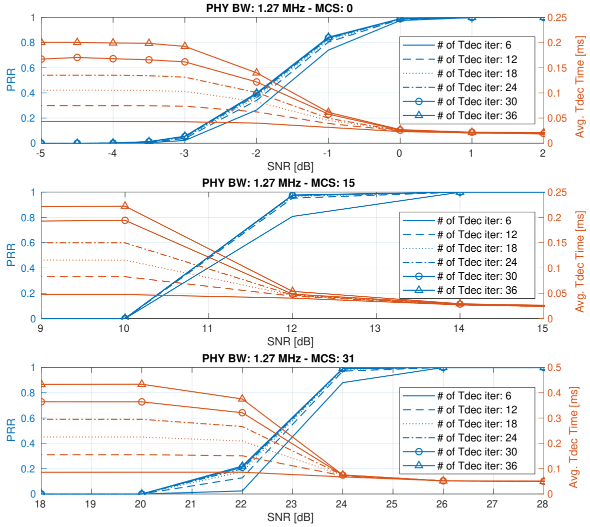


Figure 5.9: Comparison of the variation of the maximum number of turbo decoding iterations for a PHY Tx BW of 1.27 *MHz and three different MCS values.*

frames as defined in (5.5).

$$r_{\text{comb.}}(n) = \frac{1}{N_{\text{ant.}}} \sum_{i=1}^{N_{\text{ant.}}} r_{\text{Rx}_i}(n), \tag{5.5}$$

where $N_{\text{ant.}}$ is the number of Rx antennas (in our case, $N_{\text{ant.}} = 2$), $r_{\text{Rx}_i}(n)$ is the subframe synchronized at the i^{th} antenna, and $r_{\text{comb.}}(n)$ is the resulting combined subframe signal. The MCS value used in this experiment for all PHY BWs was zero. As can be seen, the Rx combining scheme employed in this experiment provided a processing gain ranging from 2.5 to 3 dB over the single Rx approach. With this scheme, SCATTER PHY was able to combine the received subframes for improved performance in low SNR scenarios.

Figure 5.9 presents the results on the variation of the maximum number of turbo decoding iterations for a PHY Tx BW of 1.27 MHz and three different MCS values, 0 (QPSK), 15 (16QAM), and 31 (64QAM). As can be seen, the PRR improved as the number of maximum iterations also increased. As can be also noticed, the PRR improvement was higher for higher MCS values, 15 and 31. This was due to the fact that as the MCS increased, the code rate increased, and consequently, the number of redundancy bits decreased, making the transmit data more prone to errors. Therefore, the probability of successful data decoding increased as the number of maximum turbo decoding iterations increased. Another important point is the trade-off between PRR improvement and the increase in decoding time. As can be verified, the PRR improved at the cost of longer decoding times for low to medium SNR values.

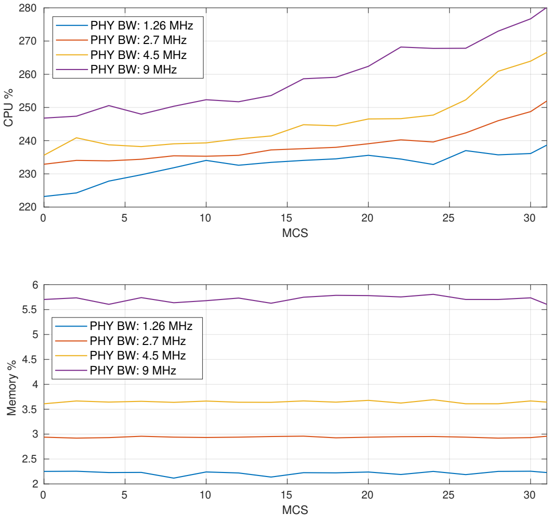


Figure 5.10: CPU and memory utilization of SCATTER PHY.

Figure 5.10 depicts the memory and CPU utilization of SCATTER PHY, using 1.27, 2.7, 4.5, and 9 MHz BWs for several MCS values. The results presented in the figure were obtained by averaging memory and CPU usage values. The memory and CPU usage values were sampled every 200 ms for the experiment's duration. In this experiment, we had SCATTER PHY operating in full-duplex mode, where one PHY transmitted to the other and vice versa. Each one of the PHYs transmitted 20 subframes in a row with a 1 ms gap between transmissions.

As can be noticed, as the selected PHY Tx BW increased, both CPU and memory utilization also increased; however, there was no memory nor CPU starvation for any of the used MCS and PHY Tx BW values. As can be also seen, for a given PHY Tx BW, as the MCS value increased, the CPU usage also increased. This was because the turbo encoding, turbo decoding, and synchronization processing tasks demanded much more CPU power for data processing as the MCS value increased, i.e., as data rates increased. For an MCS equal to 31 and a PHY Tx BW of 9 MHz, the CPU utilization of SCATTER PHY was approximately equal to 280 %. That means that less than three CPU cores were being used, consequently leaving the other cores underused for long intervals.

In the case of memory usage, it can be noticed that for a given PHY Tx BW, the memory utilization was almost constant for all MCS values, and therefore, it was practically independent of the selected MCS value. This result was expected once all the RAM memory used by SCATTER PHY was pre-allocated during its initialization. Therefore, based on the results presented in Figure 5.10, it can be concluded that SCATTER PHY did not exhaust memory or CPU resources as the configured MCS or PHY Tx BW values increased.

Figure 5.11 presents the results of the CPU consumption for the main SCAT-

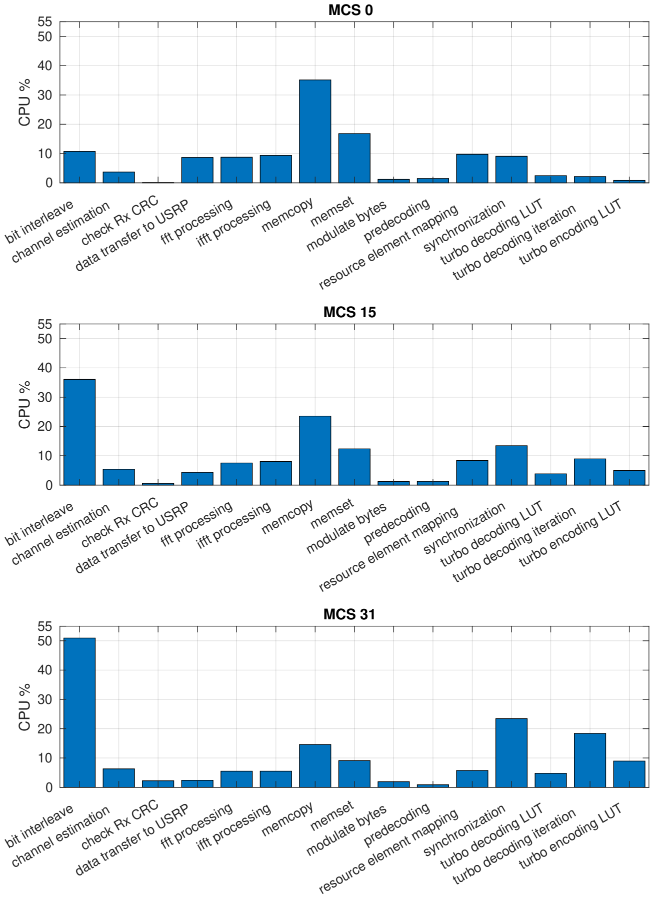


Figure 5.11: CPU profiling of SCATTER PHY for a PHY Tx BW of 4.5 *MHz.*

TER PHY's functions for three different MCS values and a PHY Tx BW of 4.5 MHz. We used valgrind with its callgrind tool for the assessment of the CPU consumption [31]. This experiment used the same setup used for the profiling of memory and CPU. The figure highlights the Rx and Tx functions with highest CPU processing load. By analyzing the figure, we see that the bit interleaving processing used more CPU as the MCS value increased. As can be seen, the CPU time-usage was approximately 11 $\%$ when the MCS value was set to zero; however, it increased to approximately 51 $\%$ when the MCS value was set to 31. The synchronization processing task required a considerable amount of CPU time for all considered MCS values. The CPU time consumption for the synchronization ranged from 9 $\%$ to more than 23 $\%$. The CPU time for the synchronization was independent of the selected MCS due to the fact that this processing did not involve data coding/decoding. It can be also noticed that the turbo decoding iteration task consumed more CPU time as the MCS increased, going from 2 % (MCS 0) to more than 18 % (MCS 31). Additionally, we see that memory copy and setting functions consumed less CPU time as the MCS value increased. This was due to the fact that other functions started consuming more CPU when the MCS increased.

5.5 Conclusions

The chapter described the design, features, and benefits of SCATTER PHY, which is an open-source physical layer that uses a filtered form of OFDM as its waveform.

The main features included using Google's Protobuf to define the message format for interaction with higher layers of the system and the ability to transmit/receive two different types of packets, which can include a synchronization signal, control data, and user data. The PHY layer communication operated with various modulation and coding schemes, several different channel BWs, and automatically improved the carrier frequency offset correction by utilizing the OFDM CP in addition to the synchronization subframe detected in the signal. SCATTER PHY minimized OOBE by utilizing custom-built FPGA based FIR filters on the transmitter side, configured for each channel bandwidth at run-time. Additionally, two independent physical interfaces were available, which effectively enabled the usage of full-duplex and theoretically double the possible throughput achieved with one interface.

The described benefits were verified experimentally. The throughput achieved with actual physical transmissions was compared to a theoretical maximum, defined by using the entire available time for communication with an effective duty cycle of 100 %. The verification showed that using full-duplex had no impact on the achieved throughput in each interface. Spectrometer measurement results showcased the capabilities of the FPGA based FIR filter, resulting in negligible OOBE, which in turn increased channel utilization and facilitated communication in neighboring channels by other radios. Using CP based CFO estimation and correction was demonstrated to outperform the sync-signal based correction in all cases, at the expense of higher processing time. Finally, the packet reception rate for all types of transmitted signals was presented, showing that SCATTER PHY could successfully decode user data at SNR values as low as 0 dB.

Acknowledgment

This work was funded by the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 732174 – Orchestration and Reconfiguration Control Architecture (ORCA) project.

References

[1] DARPA SC2. DARPA Spectrum Collaboration Challenge. Available online: Https://spectrumcollaborationchallenge.com/ (accessed on 17th of September 2019).

[2] Stojadinovic, D.; de Figueiredo, F.A.P.; Maddala, P.; Seskar, I.; Trappe, W. SC2 CIL: Evaluating the Spectrum Voxel Announcement Benefits. In Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN2019), Newark, NJ, USA, 11–14 November 2019.

[3] Giannoulis, S.; Donato, C.; Mennes, R.; de Figueiredo, F.A.P.; et al.; Dynamic and Collaborative Spectrum Sharing: The SCATTER Approach. In Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN2019), Newark, NJ, USA, 11–14 November 2019.

[4] Mennes, R.; Claeys, M.; de Figueiredo, F.A.P.; Jabandzic, I.; Moerman, I.; Latre, S. Deep Learning-Based Spectrum Prediction Collision Avoidance for Hybrid Wireless Environments. *IEEE Access* 2019, *7*, 45818–45830.

[5] Hintjens, P. *ZeroMQ Messaging for Many Applications*; O'Reilly Media, March 2013.

[6] I. Jabandžić, S. Giannoulis, R. Mennes, F. A. P. De Figueiredo, M. Claeys and I. Moerman, "A Dynamic Distributed Multi-Channel TDMA Slot Management Protocol for Ad Hoc Networks," in IEEE Access, vol. 9, pp. 61864-61886, 2021, doi: 10.1109/ACCESS.2021.3074594.

[7] R. Mennes et al., "Collaborative Flow Control in the DARPA Spectrum Collaboration Challenge," in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2024-2038, Dec. 2020, doi: 10.1109/TNSM.2020.3031078.

[8] M. Camelo et al., "An AI-Based Incumbent Protection System for Collaborative Intelligent Radio Networks," in IEEE Wireless Communications, vol. 27, no. 5, pp. 16-23, October 2020, doi: 10.1109/MWC.001.2000032.

[9] R. Mennes, F. A. P. De Figueiredo and S. Latré, "Multi-Agent Deep Learning for Multi-Channel Access in Slotted Wireless Networks," in IEEE Access, vol. 8, pp. 95032-95045, 2020, doi: 10.1109/ACCESS.2020.2995456.

[10] De Figueiredo, F.A.P.; Mennes, R.; Jiao, X.; Liu, W.; Moerman, I. A spectrum sharing framework for intelligent next-generation wireless networks. *IEEE Access* 2018, *6*, 60704–60735.

[11] De Figueiredo, F.A.P.; Stojadinovic, D.; Maddala, P.; Mennes, R.; Jabandžic, I.; Jiao, X.; Moerman, I. SCATTER PHY: An Open Source Physical Layer

for the DARPA Spectrum Collaboration Challenge. Available online: Https://github.com/zz4fap/scatter-phy (accessed on 17th of September 2019).

[12] Yu, Y.; Wang, T.; Liew, S.C. Deep-Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks. In Proceedings of the IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018.

[13] Mennes, R.; Camelo, M.; Claeys, M.; Latre, S. A Neural-Network based MF-TDMA MAC Scheduler for Collaborative Wireless Networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC),Barcelona, Spain, 15–18 April 2018.

[14] Camelo, M.; Shahid, A.; Fontaine, J.; de Figueiredo, F.A.P.; de Poorter, E.; Moerman, I.; Latre, S. A Semi-Supervised Learning Approach Towards Automatic Wireless Technology Recognition. In Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Newark, NJ, USA, 11–14 November 2019.

[15] Ta, D.T.; Nguyen-Thanh, N.; Nguyen, V.T.; Nguyen, D. Collaborative Paradigm for Next Generation Wireless Networks. submitted to *EURASIP J. Wirel. Commun. Netw.*, February 2018.

[16] Jang, S.J.; Han, C.H.; Lee, K.E.; Yoo, S.J. Reinforcement learning based dynamic band and channel selection in cognitive radio ad-hoc networks. *EURASIP J. Wirel. Commun. Netw.* 2019, *131*, doi:10.1186/s13638-019-1433-1.

[17] Ettus Research LLC. USRP Hardware Driver. Available online: Http://files.ettus.com/manual/ (accessed on 27th of August 2019).

[18] Ettus Research LLC. USRP X Series. Available online: Https://www.ettus.com/product/category/USRP-X-Series (accessed on 27th of August 2019).

[19] National Instruments. Overview of the NI USRP RIO Software-Defined Radio. White Paper; 2015. Available online: Http://www.ni.com/white-paper/52119/en/ (accessed on 27th of August 2019).

[20] Ettus Research LLC. Products. Available online: Https://www.ettus.com/product (accessed on 27th of August 2019).

[21] Google. protobuf—Protocol Buffers—Google’s Data Interchange Format. 2011. Available online: http://code.google.com/p/protobuf/ (accessed on 27th of August 2019).

[22] Gomez-Miguelez, I.; Garcia-Saavedra, A.; Sutton, P.D.; Serrano, P.; Cano, C.; Leith, D.J. srsLTE: An Open-Source Platform for LTE Evolution and Experimentation. In Proceedings of the ACM WiNTECH Workshop, New York City, NY, USA, 3–7 October 2016.

[23] Kanchi, S.; Sandilya, S.; Bhosale, D.; Pitkar, A.; Gondhalekar, M. Overview of LTE-A technology. In Proceedings of the IEEE Global High Tech Congress on Electronics (GHTCE), Shenzhen, China, 17–19 November 2014.

[24] De Figueiredo, F.A.P.; Cardoso, F.A.C.M.; Bianco F., J.A.; Vilela, R.M.; Lenzi, K.G. Multi-stage Based Cross-Correlation Peak Detection for LTE Random Access Preambles. *Rev. Telecomun.* 2013, *15*, 21–27.

[25] Chu, D.C. Polyphase codes with good periodic correlation properties. *IEEE Trans. Inf. Theory* 1972, *18*, 531–532.

[26] Sarwate, D.V.; Pursley, M.B. Crosscorrelation properties of pseudorandom and related sequences. *Proc. IEEE* 1980, *68*, 593–619.

[27] Wang, F.; Zhu, Y. An efficient CFO estimation algorithm for the downlink of 3GPP-LTE. In Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 9–11 November 2011.

[28] Bodinier, Q.; Bader, F.; Palicot, J. On Spectral Coexistence of CP-OFDM and FB-MC Waveforms in 5G Networks. *IEEE Access* 2017, *5*, 13883–13900.

[29] Lin, H.; Siohan, P. Major 5G Waveform Candidates: Overview and Comparison. In *Signal Processing for 5G*; Wiley: Hoboken, NJ, USA, 2016.

[30] Ettus Research LLC. CBX 1200-6000 MHz Rx/Tx (120 MHz, X Series Only). Available online: Https://www.ettus.com/product/details/CBX120 (accessed on 5th of September 2019).

[31] Weidendorfer, J.; Kowarschik, M.; Trinitis, C. A Tool Suite for Simulation Based Analysis of Memory Access Behavior. In Proceedings of the 4th International Conference on Computational Science (ICCS), Krakow, Poland, 6–9, June 2004.

[32] Sesia, S.; Toufik, I.; Baker, M. *LTE—The UMTS Long Term Evolution: From Theory to Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2011.

[33] Neyman, J.; Pearson, E.S. On the problem of the most efficient tests of statistical hypotheses. *Philos. Trans. R. Soc. Lond. Ser. A* 1933, *231*, 289–337.

[34] Lehtomaki, J.J.; Juntti, M.; Saarnisaari, H. *CFAR Strategies for Channelized Radiometer. IEEE Signal Process. Lett.* 2005, *12*, 13–16.

[35] Proakis, J.G. *Digital Communications*, 5th ed.; McGraw-Hill: New York, NY, USA, 2001.

[36] Papoulis, A.; Pillai, S.U. *Probability, Random Variables, and Stochastic Processes*, 4th ed.; McGraw-Hill Europe: United Kingdom, 2002.

[37] DeGroot, M.H. *Probability and Statistics*, 2nd ed.; Addison-Wesley:Massachusetts, United States, 1986.

6 Conclusion

"The only true wisdom is in knowing you know nothing."

– Socrates (470 BC - 399 BC)

This PhD work had as main goal the design and implementation of a set of advanced Software-Defined Radio (SDR) architectures and techniques for radio Hardware (HW) virtualization and efficient spectrum sharing. The results presented here shed light on and motivate for entirely new research lines towards a better understanding of virtualized and softwarized systems and also their application to disaggregated and virtualized Radio Access Network (RAN) and fairer spectrum sharing. This chapter summarizes the work that has been done in this dissertation and presents future directions that can further assist in virtualization and spectrum sharing. Next, the main contributions of this thesis are summarized.

- In Chapter 2, we designed and implemented a HW-based radio HW virtualization oriented transceiver architecture, which is meant to bring softwarization and virtualization techniques of wireless networks down to the lowest part of the wireless network stack, namely Physical (PHY) and low-Medium Access Control (MAC) layers, while maintaining high performance and low-latency as in the conventional HW-defined network. The designed architecture is implemented on a System-on-Chip (SoC) chip, which includes a Field-Programmable Gate Array (FPGA) part, and achieves efficient utilization of HW (i.e., FPGA) resources, as HW-based accelerators can be shared by multiple virtual radios. Concluding the chapter, we present a proof-of-concept demonstration that shows the feasibility of the designed architecture.

- In Chapter 3, we designed and implemented a Software (SW)-based radio HW virtualization layer that enables multiple Radio Access Technologiess (RATs) to coexist on top of the same physical Radio Frequency (RF) front-end. It is meant for multiplexing concurrent signals of the same RAT or correlated RATs. Along with the mathematical demonstration showing how the designed architecture works, we provided several simulation and experimental results attesting its efficiency and feasibility. As discussed throughout the chapter, the designed SW-based architecture has several use cases ranging from applications in Dynamic Spectrum Sharing (DSS) to network densification.
- In Chapter 4, we designed and implemented an open-source SDR-based spectrum sharing framework for intelligent next-generation wireless and mobile networks. The main goal behind this framework is to give researchers the possibility to design disruptive techniques and algorithms to maximize the sub-optimal use of the radio spectrum quickly, focusing on the methods and algorithms themselves than on a framework for experimentation. The developed framework is highly configurable and allows its functions to be easily modified or extended by simply plugging in new modules. Three specific use cases are presented with hints on how the designed framework could be employed in intelligent spectrum sharing research. To conclude the chapter, we present and discuss a handful of experimental results showing the performance and effectiveness of the designed framework.
- In Chapter 5, we introduced an open-source, highly optimized, and real-time configurable SDR-based PHY layer for the research and development of novel intelligent spectrum sharing techniques and algorithms. This PHY layer was built based on the framework introduced in Chapter 5. It has been named SCATTER PHY and was part of the SCATTER system, which was designed and built as a multi-layered system, with PHY, MAC, Artificial Intelligence (AI), and other layers. The designed PHY has the ability to communicate with the other layers through a set of well-defined messages that are exchanged over a ZeroMQ bus, which allows the other layers to access it remotely or locally. To conclude the chapter, we presented and discussed the results of several experiments, which were performed to assess the effectiveness and performance of the SCATTER PHY layer.

6.1 Future Work

Next, we present a list of possible directions for future work, related to our study and results described in this thesis, which can yield very promising research results.

- As further work regarding Chapter 2, we envision experiments of the proposed virtualization architecture with fully functional Wi-Fi, Zigbee, and

Bluetooth stacks. These experiments are essential to evaluate if the proposed architecture can cope with multiple concurrent full blown stack layers. In addition, it is still to be found where the major performance bottlenecks in our architecture are. A possible improvement to the proposed architecture that could increase resource utilization and spectrum efficiency is the use of statistical multiplexing. Therefore, the employment of this kind of multiplexing should be investigated.

- As future work regarding Chapter 3, we highlight the following improvements to the designed architecture prototype. As it was shown, channelization and Inverse Fast Fourier Transform (IFFT) (i.e., Orthogonal Frequency Division Multiplexing (OFDM) modulation) processing tasks consume a lot of Central Programming Unit (CPU) time when 12 virtual PHYs (vPHYs) are instantiated, therefore, offloading these tasks to the FPGA can increase both real-time and processing performance. Additionally, memory copy also consumes a lot of CPU time, and therefore, smarter ways of carrying out these copies should be investigated. The designed architecture supports only multi-carrier-based waveforms, however, it would be interesting to add support to other kinds of waveforms, and therefore, another direction would be adding the support to non-multi-carrier waveforms. Moreover, in our prototype, we have used a polyphase filter-bank channelizer, which divides the spectrum equally, however, in some cases, it would be interesting to have the spectrum non-equally split, therefore, the investigation of types of channelizer architectures that non-uniformly dive the spectrum should be explored in future works. Further, the integration of the designed architecture's prototype with Long-Term Evolution (LTE) or 5th Generation (5G)-like upper layers is a direct sequence of the work presented here and would serve as a demonstration of what can be achieved with the designed architecture in terms of real deployments. Moreover, in the current prototype's implementation we haven't treated OOB emissions that are brought about by the OFDM modulation. Therefore, studying and implementing techniques to reduce OOB emissions is as a possible future research direction. One could study windowing or filtering the sub-bands with a UFMC-like approach. Finally, the work presented in that chapter did not analyze the latency introduced by the designed architecture when compared to traditional architectures. Measuring the additional delay introduced by designed architecture is an interesting result that can be explored in future works.

- Regarding the work presented in Chapter 4, we can highlight further improvements and research the addition of support to 5G New-Radio (5G-NR) features like self-contained slots and scalable OFDM-based air interface. With self-contained slots, both data and Acknowledgment (ACK) information are present in the same time slot. This feature allows each transmission to be a modular transaction, giving the PHY the ability to independently decode slots and avoid static timing relationships across slots like in LTE systems. Moreover, self-contained slots are a crucial enabler to low latency.

With a scalable air interface numerology, parameters like subcarrier spacing and Transmission Time Interval (TTI) can be dynamically and efficiently modified to support the various frequency, channel Bandwidths (BWs), deployments, and services are foreseen to exist in next-generation mobile networks. With the addition of these features, the proposed framework would be similar to the 5G-NR PHY layer. Therefore, along with the RF monitor and Listen Before Talking (LBT) modules, it could be used for real-world experiments on intelligent spectrum sharing with 5G technology. The advantage of using the proposed framework for 5G experiments is that it is highly flexible and extensible by using the ZeroMQ bus, this way, it would be possible to, for example, plug an AI-driven module to it and assess the improvements such technology could provide to the spectrum optimization problem. Additionally, we also foresee the offloading of a few time-consuming PHY processing tasks (e.g., slot synchronization, Finite Impulse Response (FIR) filtering, channel coding/decoding, and OFDM modulation/demodulation) to the FPGA as a way to increase both real-time and processing performance. Moreover, the study and implementation of side-lobe suppression techniques such as cancellation carriers is a way forward into future research directions.

- The PHY layer presented in Chapter 5 has only the filtering processing offloaded onto the FPGA, however, we believe that offloading OFDM modulation and demodulation will greatly improve its performance in terms of latency and resource utilization, since CPU is released from this processing task. By offloading OFDM modulation and demodulation to the FPGA it will be possible to have symbol based filtering, which can greatly improve spectral localization and and maintain the orthogonality of the OFDM symbols. Further, interested researchers can extend the designed PHY by designing and plugging AI-based MAC modules to it and assessing the improvements on spectrum sharing.

A

Exemplary design of a SIMD FIR filter

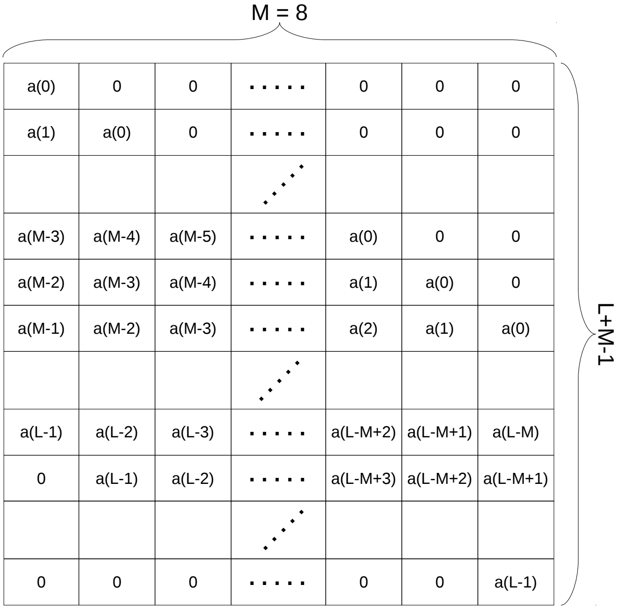
Here in this appendix, we present an example on how to use Advanced Vector Extension (AVX) instructions to implement optimized (in the sense of processing speed) FIR filters.

A.1 Design of a SIMD FIR filter

AVX are extensions to the x86 instruction set architecture for microprocessors from Intel and AMD. AVX expands most integer commands to 256 bits and therefore, the registers can store eight 32-bit single-precision floating point numbers or four 64-bit double-precision floating point numbers. We used Intel intrinsic instructions for the FIR filter development. Intel intrinsic instructions are C style functions that provide access to many Intel instructions without the need to write assembly code. A nice interactive reference tool for the intrinsic instructions can be found at [1]. FIR filters are widely employed in digital communications systems, especially in physical layer processing and here, in this work, it is mainly used to implement an filtered version of the OFDM waveform. An L-tap FIR filter can be described as

$$y(n) = \sum_{l=0}^{L-1} a(l)x(n-l), \quad 0 \leq n \leq L+N-2, \tag{A.1}$$

where $x(l)$ are the complex input samples, $y(n)$ are the complex output samples and $a(n)$ are the floating point filter coefficients. Single Instruction Multiple Data (SIMD) instructions allow the processing of multiple samples simultaneous. For



a(0)	0	0	· · · · ·	0	0	0
a(1)	a(0)	0	· · · · ·	0	0	0
			⋰			
a(M-3)	a(M-4)	a(M-5)	· · · · ·	a(0)	0	0
a(M-2)	a(M-3)	a(M-4)	· · · · ·	a(1)	a(0)	0
a(M-1)	a(M-2)	a(M-3)	· · · · ·	a(2)	a(1)	a(0)
			⋰			
a(L-1)	a(L-2)	a(L-3)	· · · · ·	a(L-M+2)	a(L-M+1)	a(L-M)
0	a(L-1)	a(L-2)	· · · · ·	a(L-M+3)	a(L-M+2)	a(L-M+1)
			⋰			
0	0	0	· · · · ·	0	0	a(L-1)

Figure A.1: Memory layout of the FIR filter kernel.

instance, SSE instructions support a 256-bit packed vector and therefore, it can process 8 single-precision or 4 double-precision floating values at the same time. Therefore, with AVX functions it is possible to perform $M = 8$ operations at the same time. An important hint on the operation of the intrinsics instructions is that all the data should be aligned for improved performance of the instructions. When using AVX intrinsics, all data should be aligned to a 32-byte memory boundary, i.e., the memory address must be divisible by 32.

In order to speed up the filter processing, the data layout in memory of the filter coefficients should be meticulously designed. Figure A.1 depicts the memory layout of the filter kernel (i.e., the FIR filter coefficients). Each row of the filter kernel forms a vector containing M elements for the AVX operations. As can be seen in the figure, the FIR filter coefficients are repeated in each column by applying an offset of one element to each new column. Therefore, the filter kernel is a matrix with dimension $(L + M - 1) \times M$. The filter coefficients are real numbers represented by 1 single-precision floating point value.

The Algorithm 1 shows a exemplary pseudo-code on how a FIR filter can be implemented to take advantage of the AVX functions, and consequently speed-up the filtering time. It receives an input complex array of length N, the filter kernel, which is composed of $(L + M - 1)$ packed vectors with M elements each (i.e., each row is a 256-bit packed vector), and outputs the filtered signal in the output complex array, which will have a length of $L + N - 1$. We assume that the length of the input signal vector is an integer multiple of M, otherwise, it is necessary to

Table A.1: FIR filter processing time for different implementation versions.

FIR order	FIR version	Time [ms]	Speed-up %
64	naive	0.954478	-
	no unrolling	0.133955	85.966
	unrolling	0.093086	90.247
128	naive	1.986	-
	no unrolling	0.234	88.211
	unrolling	0.197	90.090

pad zeros to the vector until its length becomes an integer multiple of M. Each sample of the input signal is a complex number represented by 2 single-precision floating point values, i.e., real (In-phase) and imaginary (Quadrature) values. This way, when a complex sample vector is cast into a floating-point vector (see lines 11 and 12), the even and odd indexes represent real (I) and imaginary (Q) values respectively.

In the outer loop, after casting the complex input vector into a floating-point vector, in lines 17 to 28, 16 consecutive floating-point values are rearranged into two 256-bit packed vectors, the first one containing only real values and the second one containing only imaginary values. At each iteration of the outer loop, 8 real and 8 imaginary values are loaded into 256-bit packed vectors. In the inner loop, the the dot product between each one of the rows of the filter kernel and the real packed vector is calculated from lines 33 to 39 and the result is accumulated to the even indexed output vector, in line 50. The the dot product between each one of the rows of the filter kernel and the imaginary packed vector is calculated from lines 41 to 47 and the result is accumulated to the add indexed output vector, in line 51. Therefore, at each iteration of the inner loop two, real and imaginary, output values are calculated and accumulated in the output vector. The output complex vector should be zeroed before the filtering as we use its contents to accumulate the output values.

It is worth mentioning that the filter processing time can be further decreased by unrolling the inner-loop, which is done by replicating lines 33 to 52 for subsequent rows of the filter kernel.

In Table A.1 we compare the processing time for different versions of the FIR filter implementation. For this comparison we use an input signal with 5760 complex samples and FIR filters with length equal to 65 and 129 respectively, i.e., 64-th and 128-th order FIR filters. The FIR processing time is averaged over 100×10^3 repetitions of the filtering operation. The *naive* version corresponds to the straight-forward implementation of (A.1), the *no unrolling* version corresponds to the pseudo-code presented in Algorithm 1 and the *unrolling* version corresponds to unrolling the inner-loop 8 times.

algorithm 1: Pseudo-code for an AVX optimized FIR filter.

```
procedure AVXFIRFILTER(cf_t *cplx_in, int in_len, __m256 *filter_kernel, int filter_len, cf_t *cplx_out)
    __attribute__((aligned(32))) __m256 blk0, blk1, blkI, blkQ;
    __attribute__((aligned(32))) __m256 dpI, resI, dpQ, resQ;

    ▷ Static initializations.
    __m256i mask = _mm256_setr_epi32(0, 1, 4, 5, 2, 3, 6, 7);
    const int imm8_I = _MM_SHUFFLE(2, 0, 2, 0);
    const int imm8_Q = _MM_SHUFFLE(3, 1, 3, 1);
    ▷ Cast complex samples into single precision floating-point type.
    ▷ Note that one complex sample becomes 2 floating-point values.
    __attribute__((aligned (32))) float *in = (float*)cplx_in;
    __attribute__((aligned (32))) float *out = (float*)cplx_out;
    int cnt = 0, nof_outer_loop_iter = in_len/M;

    ▷ Outer loop.
    for $k \leftarrow 0$ $to$ (nof_outer_loop_iter - 1) do
        ▷ Load 8 values from the input vector.
        blk0 = _mm256_load_ps(in+(k*2*M));
        ▷ Load the next 8 values from the input vector.
        blk1 = _mm256_load_ps(in+(k*2*M)+M);
        ▷ Get 8 real values in vector. They are not in time sequence.
        blkI = _mm256_shuffle_ps(blk0, blk1, imm8_I);
        ▷ Organize real samples in time sequence.
        blkI = _mm256_permutevar8x32_ps(blkI, mask);
        ▷ Get 8 imag values in vector. They are not in time sequence.
        blkQ = _mm256_shuffle_ps(blk0, blk1, imm8_Q);
        ▷ Organize imag samples in time sequence.
        blkQ = _mm256_permutevar8x32_ps(blkQ, mask);

        cnt = 0;
        ▷ Inner loop.
        for $i \leftarrow 0$ $to$ (filter_len + M - 2) do
            ▷ Multiply lower 4 real elements and sum their products.
            ▷ Simulaneoulsy does the same for the higher 4 elements.
            dpI = _mm256_dp_ps(filter_kernel[i], blkI, 0xff);
            ▷ Swap lower 4 elements with higher 4 elements.
            resI = _mm256_permute2f128_ps(dpI, dpI, 1);
            ▷ Get the dot product of 2 × eight elements.
            resI = _mm256_add_ps(resI, dpI);

            ▷ Multiply lower 4 imag elements and sum their products.
            ▷ Simulaneoulsy does the same for the higher 4 elements.
            dpQ = _mm256_dp_ps(filter_kernel[i], blkQ, 0xff);
            ▷ Swap lower 4 elements with higher 4 elements.
            resQ = _mm256_permute2f128_ps(dpQ, dpQ, 1);
            ▷ Get the dot product of 2 × eight elements.
            resQ = _mm256_add_ps(resQ, dpQ);

            ▷ Accumulates the results in the output vector.
            out[k*2*M + cnt] += resI[0];
            out[k*2*M + cnt + 1] += resQ[0];
            cnt += 2;
        end for
    end for
end procedure
```

References

[1] Intel Corporation, *Intel's Intrinsics Guide*, [online]. Available: https://software.intel.com/sites/landingpage/IntrinsicsGuide/

B

Design of a sub-band Occupancy Algorithm

In this appendix, we describe the sub-band Occupancy Algorithm in details.

B.1 sub-band Occupancy Algorithm

As mentioned earlier, the algorithm is based on the Cell-Average Constant False Alarm Rate (CA-CFAR) algorithm [1]. It is employed for detecting the presence of transmissions, i.e., occupancy, of subbands composing the bandwidth being assessed. The algorithm is used to calculate an occupancy (i.e., detection) threshold based on a reference set of sub-band power values. The algorithm can be split into two three stages: (i) removal of samples that are not considered as only being noise observations, this step is also known as censoring, (ii) calculation of a noise-power threshold, and (iii) the actual sub-band occupancy detection. This procedure makes it possible to reliably decide whether a sub-band, sb, is already being occupied by another user or not. Figure B.1 summarizes the three stages composing the sub-band Occupancy algorithm.

The sample power calculation and sub-band grouping processing tasks are realized as showed in Figure 4.10 and described in Section 4.5. The subsequent signal processing is performed in three stages making use of the sub-band power samples. The sub-band occupancy procedure consists basically of hypothesis testing following the Neyman&Pearson lemma [2]. This lemma establishes that sub-band

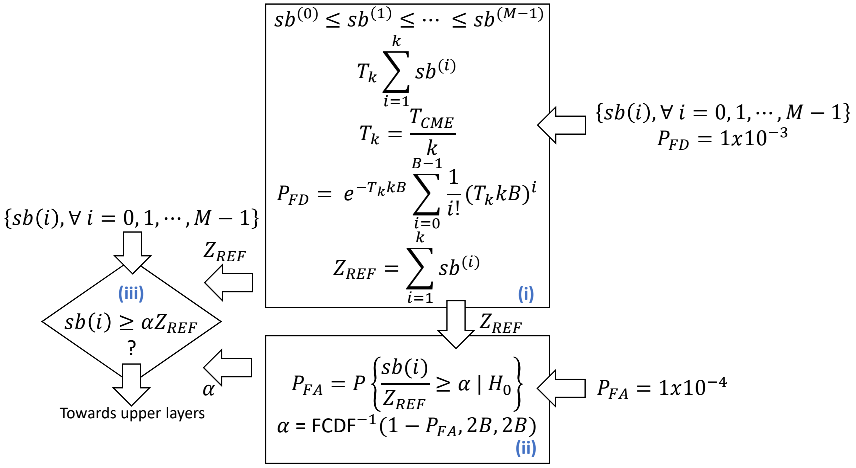

$sb^{(0)} \leq sb^{(1)} \leq \cdots \leq sb^{(M-1)}$
$T_k \sum_{i=1}^{k} sb^{(i)}$
$T_k = \frac{T_{CME}}{k}$
$P_{FD} = e^{-T_k kB} \sum_{i=0}^{B-1} \frac{1}{i!}(T_k kB)^i$
$Z_{REF} = \sum_{i=1}^{k} sb^{(i)}$
(i)
$\{sb(i), \forall\, i = 0, 1, \cdots, M-1\}$
$P_{FD} = 1x10^{-3}$
$\{sb(i), \forall\, i = 0, 1, \cdots, M-1\}$
Z_{REF}
(iii)
$sb(i) \geq \alpha Z_{REF}$?
α
Towards upper layers
Z_{REF}
$P_{FA} = P\left\{\frac{sb(i)}{Z_{REF}} \geq \alpha \mid H_0\right\}$
α = FCDF$^{-1}(1 - P_{FA}, 2B, 2B)$
(ii)
$P_{FA} = 1x10^{-4}$

Figure B.1: Summary of the sub-band occupancy algorithm: stage (i) is used to calculate the noise-power reference, Z_{REF}; stage (ii) is used to calculate the threshold, α; and stage (iii) applies hypothesis test to all the sub-band power values, $sb(i)$.

occupancy testing is based on the following likelihood ratio inequality:

$$\frac{sb^{H_1}}{sb^{H_0}} > \alpha, \tag{B.1}$$

where the hypothesis H_0 is rejected in favor of H_1 when the sub-band contains not only noise, i.e., the presence of a transmission. This hypothesis testing is optimum only when the cumulative distribution function (CDF) of the ratio in (B.1) given the hypothesis H_0 is known, so that it is possible to calculate a threshold value that satisfies:

$$P\left\{\frac{sb^{H_1}}{sb^{H_0}} > \alpha \mid H_0\right\} = P_{FR}, \tag{B.2}$$

for a given probability of false rejection, P_{FR}. Therefore, normally, the derivation of the threshold assumes knowledge of the probability distribution function (PDF) of both random variables sb^{H_1} and sb^{H_0}.

B.1.1 Censoring and Noise Power Reference, Z_{REF} Calculation Stage

In this stage, corrupted sub-band power samples are removed from the reference set. Corrupted sub-band power samples are caused by the presence of transmissions in that section (sub-band) of the assessed spectrum band. For the censoring of sub-band power samples, we use the Forward Consecutive Mean Excision (FCME) algorithm [3]. Initially, the values in the sub-band power reference set, defined as

$$\{sb(l) \mid l = 0, 1, \cdots, M-1\}, \tag{B.3}$$

are sorted in ascending order, so that

$$\left\{sb^{(i)} \mid i = 0, 1, \cdots, M-1\right\}, \tag{B.4}$$

where

$$sb^{(0)} \leq sb^{(1)} \leq sb^{(2)} \leq \cdots \leq sb^{(M-1)}. \tag{B.5}$$

The censoring at the k-th step is performed with the following inequality

$$sb^{(k+1)} \geq T_k \sum_{i=1}^{k} sb^{(i)}, \tag{B.6}$$

where T_k is the censoring scaling factor at the k-th step. If the inequality is satisfied (i.e., true), all the greater values are decided to be corrupted by the presence/occupancy of transmissions/interference from other users. The test of (B.6) is performed iteratively, being necessary to calculate the censoring scaling factor, T_k, for each new iteration. The censoring scaling factor, T_k, controls the properties of the censoring stage, and it is normally chosen so that the probability of false disposal, P_{FD}, has a pre-defined value. P_{FD} is the probability of making a incorrect decision during the k-th step of the censoring stage. The initial T_k value is calculated assuming that $sb^{(k+1)}$ is a sub-band power sample that only contains noise observation, i.e., it is free of transmission/interference signals. Under this assumption, the probability that the inequality in (B.6) is true corresponds to a probability of false disposal, P_{FD} defined by

$$P_{FD} = P\left\{sb^{(k+1)} \geq T_k \sum_{i=1}^{k} sb^{(i)} \mid H_0\right\}. \tag{B.7}$$

Each iteration of the censoring stage starts with k equal to the smallest assumed clean set of subbands samples. The larger the smallest assumed clean set of subbands samples, the better the censoring algorithm works [3]. However, the larger the assumed clean set of subbands samples, the higher is the probability that corrupted sub-band samples will be part of the initial clean set. The testing of (B.6) continues until it is true for some value of k or all reference sub-band power samples are decided to be unoccupied, i.e., free of transmissions/interference.

Considering the H_0 hypothesis (i.e., only noise observations) and that noise follows a complex Gaussian distribution, $\mathcal{CN}(0, \sigma^2)$, the noise distribution after the N-point Fast Fourier Transform (FFT) also follows a complex Gaussian distribution (the only difference is a scaling factor, N, due to the FFT processing), defined by $\mathcal{CN}(0, N\sigma^2)$. Therefore, each one of the sub-band power values, $sb(i) = \sum_{b=0}^{B-1} |x[b]|^2, i = 0, 1, \cdots, M-1$, follows a Gamma distribution, $\Gamma(B, N\sigma^2)$ with mean given by

$$\mathbb{E}[sb^{(\cdot)}] = BN\sigma^2. \tag{B.8}$$

Therefore, since that

$$\frac{\sum_{i=0}^{k-1} sb^{(i)}}{k} \approx \mathbb{E}[sb^{(\cdot)}] = BN\sigma^2, \tag{B.9}$$

the equation for the probability of false disposal, P_{FD}, can be approximated as

$$P_{FD} \approx P\left\{sb^{(k+1)} \geq T_k k B N \sigma^2\right\}. \tag{B.10}$$

This approximation becomes better as the number of reference sub-band power samples, k, increases. Additionally, given that $sb^{(k+1)}$ is a Gamma random variable, $\Gamma(k, \theta)$, with shape parameter being a positive integer number, B, its CDF can be simplified and expressed in closed form as [4, 5]

$$F(y, k, \theta) = P(Y \leq y) = 1 - e^{-y/\theta} \sum_{i=0}^{k-1} \frac{1}{i!} \left(\frac{y}{\theta}\right)^i, \tag{B.11}$$

where k and θ are the shape and scale parameters respectively.

Therefore, the probability of false discard (or censoring) in (B.10) can be calculated as

$$\begin{aligned} P_{FD} &= 1 - P\left\{sb^{(k+1)} \leq T_k k B N \sigma^2\right\} \\ &= e^{-T_k k B} \sum_{i=0}^{B-1} \frac{1}{i!} \left(T_k k B\right)^i, \end{aligned} \tag{B.12}$$

where in (B.11), y was made equal to $T_k k B N \sigma^2$. Here we define $T_{CME} = kT_k$, which represents a input parameter for the FCME algorithm and can be found by solving (B.12). P_{FD} can be understood as the desired clean sample rejection rate.

The censoring threshold is defined as $T_k \sum_{i=1}^{k} sb^{(i)}$ and therefore, sub-band power samples that have value greater or equal than the threshold are censored (i.e., discarded) and the new set of sub-band power samples consists of the remaining samples, which are assumed to contain only noise observations. The output of the disposal, i.e., censoring, stage are composed by the number of clean sub-band power samples, k, and the *noise-power reference*, Z_{REF}, which is defined by the summation of all sub-band power samples that have value below the censoring threshold. Z_{REF} is defined as

$$Z_{REF} = \sum_{i=1}^{k} sb^{(i)}. \tag{B.13}$$

B.1.2 Threshold Calculation Stage

After selecting the set of sub-band power samples that are considered as only being noise observations, the next stage consists of calculating the occupancy (or detection) threshold, α. The occupancy threshold, α, is calculated according to the employed decision method, which here is selected as being the Cell Averaging (CA) method [3].

The occupancy threshold, α, is calculated under the assumption that there is no transmission/interference present in the reference sub-band power samples for a given probability of false alarm, P_{FA}, which is defined as

$$P_{FA} = P\left\{ \frac{sb(i)}{Z_{REF}} \geq \alpha \mid H_0 \right\}. \tag{B.14}$$

As the quadrature components (i.e., complex samples) at the output of the FFT are assumed to follow the complex Gaussian distribution, $\mathcal{CN}(0, N\sigma^2)$, it can be consequently assumed that the sub-band power samples $sb(.)$ and the noise-power reference, Z_{REF}, follow scaled central Chi-squared distributions, $\frac{N\sigma^2}{2}\chi^2(2B)$, $\frac{N\sigma^2}{2}\chi^2(2Bk)$ with $2B$ and $2Bk$ degrees of freedom, respectively. Additionally, it is known that the ratio between two central Chi-squared random variables results in a random variable that follows a *Fisher* distribution [3, 6], whose CDF is defined as

$$\text{FCDF}(\alpha) = 1 - P\left\{ \frac{sb(i)/2B}{Z_{REF}/2Bk} \geq k\alpha \right\}, \tag{B.15}$$

therefore, the occupancy threshold, α, is calculated as

$$\alpha = \text{FCDF}^{-1}(1 - P_{FA}, 2B, 2Bk)/k. \tag{B.16}$$

where FCDF is the *Fisher* CDF. Note that the scaling factor, $\frac{N\sigma^2}{2}$, is the same for $sb(.)$ and Z_{REF}, and therefore, it disappears when the ratio is taken.

B.1.3 sub-band Occupancy Testing Stage

After calculating the occupancy threshold, α, and the noise-power reference, Z_{REF}, the final decision if a sub-band is free or occupied is made by evaluating the following hypothesis test

$$sb(i) \geq \alpha Z_{REF}, \forall\, i = 0, 1, \cdots, M-1. \tag{B.17}$$

If the test is true, signal(s)-plus noise hypothesis H_1 is chosen, i.e., the sub-band is occupied by a transmission. Otherwise, the noise-only hypothesis H_0 is decided to be true and the sub-band is declared unoccupied.

References

[1] Felipe A. P. de Figueiredo, Fabbryccio A. C. M. Cardoso, Jose A. Bianco F., Rafael M. Vilela, Karlo G. Lenzi, *Multi-stage Based Cross-Correlation Peak Detection for LTE Random Access Preambles*, Revista Telecomunicacoes, vol. 15, September 2013.

[2] J. Neyman and E. S. Pearson, *On the problem of the most efficient tests of statistical hypotheses*, Philosophical Transactions of the Royal Society of London, Series A. 231: 289-337, 1933.

[3] Janne J. Lehtomaki, Markku Juntti and Harri Saarnisaari, *CFAR Strategies for Channelized Radiometer*, IEEE Signal Processing Letters, vol. 12, no. 1, January, 2005.

[4] J. G. Proakis, *Digital Communications*, 5th Edition, McGraw-Hill, New York, 2001.

[5] Papoulis, Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th Edition, McGraw-Hill Europe, January 2002.

[6] M. H. DeGroot, *Probability and Statistics*, 2nd Edition, Addison-Wesley, 1986.

C
Design of a Two-Stage Detection Algorithm

In this appendix, we describe in detail a two-stage synchronization sequence detection algorithm.

C.1 Two-Stage Detection Algorithm

The very first processing executed by SCATTER PHY is the detection of incoming synchronization subframes, which might be followed by data-only subframes. This is a task of utmost importance once the correct detection of subframes is crucial in establishing links in scenarios with low Signal to Noise Ratio (SNR) or limited by interference. In this Appendix, we describe the proposed detection algorithm employed in SCATTER PHY.

For improved performance, the proposed algorithm was split into two stages. In the first stage, a buffer containing received samples with the length of a subframe was correlated with a local version of the Synchronization (Synch) subframe, and then, if the Peak-to-Side-lobe Ratio (PSR) was greater than or equal to a predefined threshold, $\text{Thresh}_{\text{1st\_stage}}$, then the detection algorithm proceeded to the second stage. The Power Delay Profile (PDP) between the received signal and the local version of the sync subframe containing only the synchronization signal is defined as:

$$\text{PDP}_{\text{1st\_stage}}(l) = \left| \sum_{n=0}^{N_{\text{S}}-1} r[n] r_l^*[(n+l)_{N_{\text{S}}}] \right|^2, \quad 0 \leq l \leq N_{\text{S}} - 1, \tag{C.1}$$

where $r(n)$ is the received signal, $r_l(n)$ is the local version Synch subframe, N_{S} is the length of a subframe, $(.)_{N_{\text{S}}}$ denotes that this is a cyclic correlation, and $(.)^*$ is the complex conjugate. Therefore, the detection algorithm proceeds to the second stage if the PSR of the first stage is greater than or equal to the first stage threshold, as defined in (C.2). Figure C.1 summarizes the processing carried out in the first stage of the synchronization sequence detection algorithm. As is seen in the figure, we used frequency domain cross-correlation as its implementation with FFTs and an IFFT is faster than the time domain version of the correlation [1].

$$\text{PSR}_{\text{1st\_stage}} = \frac{\arg\max_l \text{PDP}_{\text{1st\_stage}}(l)}{\arg\max_{i, i \neq l} \text{PDP}_{\text{1st\_stage}}(l)} \geq \text{Thresh}_{\text{1st\_stage}}. \tag{C.2}$$

In the second stage, if $\text{PSR}_{\text{1st\_stage}} \geq \text{Thresh}_{\text{1st\_stage}}$, then the l^{th} lag that maximized the numerator in (C.2) possibly points to the middle of the synchronization subframe. Based on the position of the l^{th} lag, it is possible to extract (i.e., convert the Synch OFDM symbol into frequency domain and extract the 72 complex symbols carrying the Synch sequence) the 72 symbol long synchronization sequence and correlate it with its local version as defined next. The PDP between the received Synch sequence and its local version is defined as:

$$\text{PDP}_{\text{2nd\_stage}}(m) = \left| \sum_{n=0}^{N_{\text{Synch}}-1} S[n] S_l^*[(n+m)_{N_{\text{Synch}}}] \right|^2, \quad 0 \leq m \leq N_{\text{Synch}} - 1, \tag{C.3}$$

where $S(n)$ is the received Synch sequence, $S_l(n)$ is the local version of the Synch sequence, N_{Synch} is the length of the Synch sequence, $(.)_{N_{\text{Synch}}}$ denotes that this is a cyclic correlation, and $(.)^*$ is the complex conjugate. The signal processing carried out at the second stage is based on $\text{PDP}_{\text{2nd\_stage}}(m)$. For the sake of brevity, we use $\text{PDP}(m)$ to denote $\text{PDP}_{\text{2nd\_stage}}(m)$ in what follows.

The second stage of the proposed detection algorithm is based on the CA-CFAR algorithm [2]. It is used to detect the existence of the synchronization sequence within a given received signal buffer. The algorithm is used to calculate a variable detection threshold, which varies according to the noise power, based on the second stage PDP set of samples, $\text{PDP}_{\text{2nd\_stage}}(m)$. The algorithm is divided into three main steps: (i) elimination (i.e., also referred as censoring) of samples that are not regarded as being noise-only (i.e., pure noise samples) observations, (ii) computation of the noise-power threshold level, and finally, (iii) the hypothesis test used to detect the synchronization sequence. These steps make it possible to

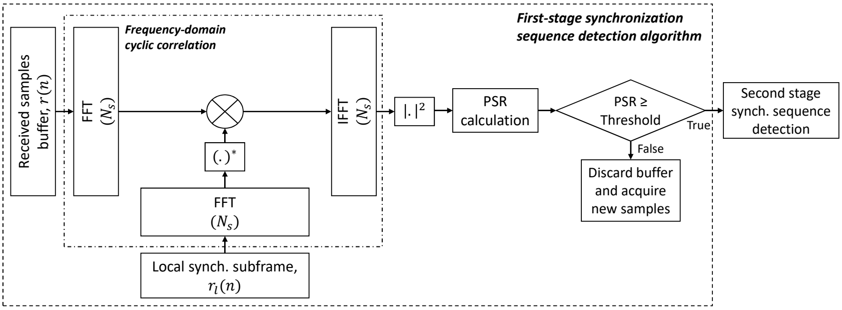

Received samples buffer, $r(n)$
FFT (N_s)
Frequency-domain cyclic correlation
(.)*
FFT (N_s)
IFFT (N_s)
Local synch. subframe, $r_l(n)$
|.|²
PSR calculation
PSR ≥ Threshold
True
False
Discard buffer and acquire new samples
First-stage synchronization sequence detection algorithm
Second stage synch. sequence detection

Figure C.1: Summary of the first-stage synchronization sequence detection algorithm.

decide reliably whether the synchronization sequence is present or not in a given received buffer. Figure C.2 summarizes the three steps composing the Synch sequence detection algorithm employed in the second stage. Next, we describe each one of the three steps composing the second stage Synch detection algorithm.

The procedure followed in the second-stage of the synchronization sequence detection algorithm essentially involved a hypothesis test that followed the Neyman and Pearson lemma described in [3]. The Neyman and Pearson lemma defines that the hypothesis testing used to detect the presence of synchronization sequence is based on the inequality shown next.

$$\frac{\text{PDP}_{H_1}}{\text{PDP}_{H_0}} > \alpha, \tag{C.4}$$

where the hypothesis H_0 is discarded (i.e., rejected) in favor of the hypothesis H_1 when the PDP set contains the desired signal, i.e., the synchronization sequence. Therefore, the hypothesis H_0 is the one representing the case where only noise is present. The just described hypothesis test is said to be optimum when and only when the CDF of the ratio in (C.4) gives that the hypothesis H_0 is known, so that it is possible to compute a threshold level that fulfill the following equality:

$$P\left\{\frac{\text{PDP}_{H_1}}{\text{PDP}_{H_0}} > \alpha \mid H_0\right\} = P_{FR}, \tag{C.5}$$

for a pre-defined (i.e., given) probability of false-rejection, P_{FR}. Therefore, the calculation of the detection threshold level considers that the PDF of both random variables, PDP_{H_1} and PDP_{H_0}, is known.

C.1.1 Step # 1: Censoring and Calculation of the Noise-Power Reference Level

In this first step, PDP samples containing the presence of other than noise observations, i.e., corrupted PDP samples, are removed (i.e., censored) from the PDP reference set. Corrupted samples are caused by the presence of samples containing

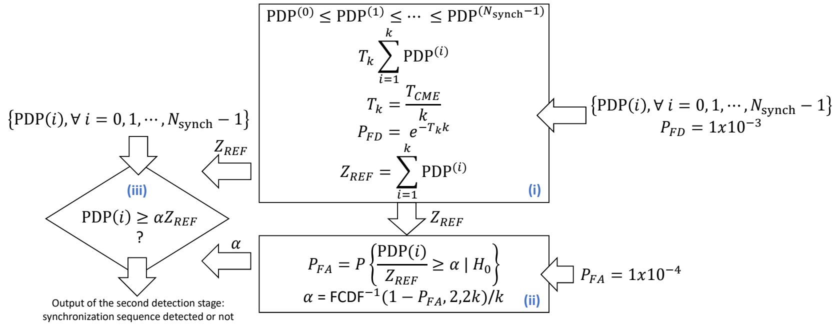

PDP^(0) ≤ PDP^(1) ≤ ⋯ ≤ PDP^(N_synch−1)
T_k Σ_{i=1}^{k} PDP^(i)
T_k = T_CME / k
P_FD = e^(−T_k k)
Z_REF = Σ_{i=1}^{k} PDP^(i)
(i)
{PDP(i), ∀ i = 0, 1, ⋯, N_synch − 1}
P_FD = 1x10^−3
Z_REF
P_FA = P{PDP(i) / Z_REF ≥ α | H_0}
α = FCDF^−1(1 − P_FA, 2,2k)/k
(ii)
P_FA = 1x10^−4
{PDP(i), ∀ i = 0, 1, ⋯, N_synch − 1}
Z_REF
(iii)
PDP(i) ≥ αZ_REF
?
α
Output of the second detection stage:
synchronization sequence detected or not

Figure C.2: Summary of the second-stage Synch sequence detection algorithm: Step (i) is employed to calculate a reference for the noise-power, Z_{REF}; Step (ii) is employed to calculate the detection-threshold, α; and Step (iii) applies a test of hypothesis to the values of the second stage PDP.

noise plus another signal (i.e., a signal different from noise). For the discarding/removal of corrupted PDP samples, we employed the algorithm known as FCME [4]. To begin with, the samples in the PDP set can be defined as:

$$\{\text{PDP}(m) \mid m = 0, 1, \cdots, N_{\text{Synch}} - 1\}, \tag{C.6}$$

and then, the PDP samples are sorted in ascending order of values; this way, the sorted PDP set can be defined as:

$$\left\{\text{PDP}^{(i)} \mid i = 0, 1, \cdots, N_{\text{Synch}} - 1\right\}, \tag{C.7}$$

where:

$$\text{PDP}^{(0)} \leq \text{PDP}^{(1)} \leq \text{PDP}^{(2)} \leq \cdots \leq \text{PDP}^{(N_{\text{Synch}}-1)}. \tag{C.8}$$

The discarding of corrupted PDP samples, at the k^{th} iteration of the algorithm, is carried out through the application of the following inequality:

$$\text{PDP}^{(k+1)} \geq T_k \sum_{i=1}^{k} \text{PDP}^{(i)}, \tag{C.9}$$

where T_k corresponds to the discarding/censoring scaling factor at the k^{th} iteration of the censoring algorithm. If the inequality shown in (C.9) holds, all the PDP samples that present values greater than PDP$^{(k+1)}$ are considered to be corrupted. The testing of (C.9) is carried out iteratively, where it is necessary to compute the censoring scaling factor, T_k, for every new iteration of the algorithm. The discarding scaling factor, T_k, rules the properties of the discarding step of the detection algorithm. This parameter is generally chosen in order that the probability of false disposal, P_{FD}, presents a pre-defined value. The probability of making an incorrect decision during the k^{th} iteration of the discarding step is denoted by P_{FD}. The initial value of T_k is computed considering that PDP$^{(k+1)}$ is a PDP sample that does not contain the presence of the synchronization sequence, i.e., a PDP sample containing a noise-only observation. Based on this premise, the probability that the inequality given by (C.9) holds corresponds to the probability of false disposal, P_{FD}, defined as:

$$P_{FD} = P\left\{ \text{PDP}^{(k+1)} \geq T_k \sum_{i=1}^{k} \text{PDP}^{(i)} \mid H_0 \right\}. \tag{C.10}$$

Every iteration of the discarding step initializes with the parameter k being equal to the smallest supposed clean set (i.e., noise-only set) of PDP samples. The larger the supposed clean set, the better the discarding algorithm works [4]. However, the larger the supposed clean set is, the higher is the likelihood that corrupted PDP samples (i.e., samples containing other than noisy-only observations) will be part of the initial supposed clean set. The testing of (C.9) proceeds until it is true for some value of the parameter k or all PDP samples used as a reference are considered to be noise-only (i.e., signal-free) samples.

Considering the H_0 hypothesis (i.e., observation of samples only containing noise) and that the noise is a stochastic processes following a complex Gaussian distribution, $\mathcal{CN}(0, \sigma^2)$, therefore, each one of the PDP values follows a Gamma distribution, $\Gamma(1, N_{\text{Synch}}\sigma^2)$ with mean given by:

$$\mathbb{E}[\text{PDP}^{(\cdot)}] = N_{\text{Synch}}\sigma^2. \tag{C.11}$$

Therefore, since:

$$\frac{\sum_{i=0}^{k-1} \text{PDP}^{(i)}}{k} \approx \mathbb{E}[\text{PDP}^{(\cdot)}] = N_{\text{Synch}}\sigma^2, \tag{C.12}$$

the equation defining the probability of false disposal, P_{FD}, can be accurately approximated as:

$$P_{FD} \approx P\left\{ \text{PDP}^{(k+1)} \geq T_k k N_{\text{Synch}}\sigma^2 \right\}. \tag{C.13}$$

The approximation given by (C.13) becomes more and more accurate as the amount of reference PDP samples, i.e., k, increases. Moreover, given that the PDP sample PDP$^{(k+1)}$ is a random variable with Gamma distribution, $\Gamma(k, \theta)$, with the shape parameter, k, equal to one, therefore, as demonstrated in [5, 6], its CDF can be simplified and expressed in closed-form by:

$$F(y, k, \theta) = P(Y \leq y) = 1 - e^{-y/\theta} \sum_{i=0}^{k-1} \frac{1}{i!} \left(\frac{y}{\theta} \right)^i, \tag{C.14}$$

where θ and k are the scale and shape parameters of the Gamma distribution, respectively.

Consequently, the probability of false censoring given by (C.13) can be computed as:

$$\begin{aligned} P_{FD} = 1 - P\left\{\text{PDP}^{(k+1)} \leq T_k k N_{\text{Synch}}\sigma^2\right\} \\ = e^{-T_k k}, \end{aligned} \tag{C.15}$$

where in (C.14), y has been set to $T_k k N_{\text{Synch}}\sigma^2$. Next, we establish that $T_{CME} = kT_k$, where T_{CME} is an input parameter to the FCME algorithm and is computed by solving (C.15). Therefore, the probability of false censoring, P_{FD}, can be thought of as the desired rejection rate to obtain a clean sample set.

The censoring/discarding threshold level is defined as $T_k \sum_{i=1}^{k} \text{PDP}^{(i)}$. This way, PDP samples that present values greater than or equal to the threshold level are censored (i.e., removed from the set). Next, the new set of samples consists of samples that are considered to be noise-only observations. The outcome of the discarding step consists of (i) the noise-power reference level, Z_{REF}, and (ii) the number of noise-only samples, k. The noise-power reference level is computed as the summation of all PDP samples that present values that are less than the discarding threshold level. The noise-power reference level, Z_{REF}, is given by:

$$Z_{\text{REF}} = \sum_{i=1}^{k} \text{PDP}^{(i)}. \tag{C.16}$$

C.1.2 Step # 2: Calculation of the Detection Threshold Level

Following the computation of the noisy-only set of PDP samples, in Step $\#1$ of the algorithm, the next step of the algorithm involves computing the synchronization detection threshold level, α. The synchronization detection threshold level, α, is computed following the decision method known in the literature as the CA method [4].

The synchronization detection threshold level, α, is computed assuming that there is no synchronization sequence present within the set of PDP reference samples for a given probability of false alarm, P_{FA}. The probability of false alarm, P_{FA}, is given by:

$$P_{FA} = P\left\{\frac{\text{PDP}(i)}{Z_{\text{REF}}} \geq \alpha \mid H_0\right\}. \tag{C.17}$$

Next, we consider that the noise-power reference, Z_{REF}, and the PDP samples both follow scaled central Chi-squared distributions, given by $\frac{N_{\text{Synch}}\sigma^2}{2}\chi^2(2k)$ and $\frac{N_{\text{Synch}}\sigma^2}{2}\chi^2(2)$ with $2k$ and 2 degrees of freedom, respectively. Moreover, it is known from the theory of random variables that the ratio of two central Chi-squared random variables results in a random variable that exhibits a Fisher distribution [4, 7], whose CDF is given by:

$$\text{FCDF}(\alpha) = 1 - P\left\{\frac{\text{PDP}(i)/2}{Z_{\text{REF}}/2k} \geq k\alpha\right\}, \tag{C.18}$$

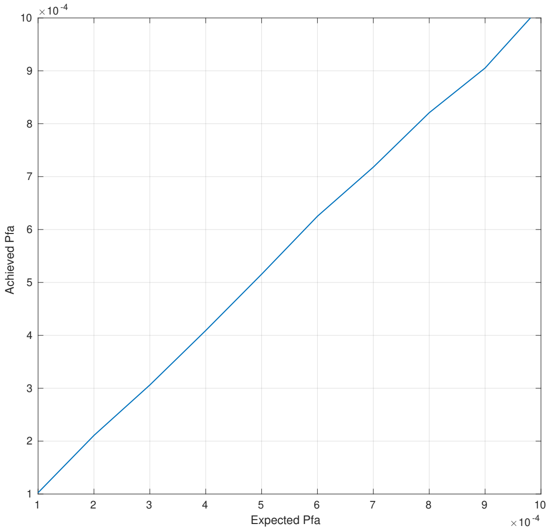


Figure C.3: Comparison between the expected P_{FA} and the actual value for an SNR of 0 dB.

consequently, the synchronization detection threshold level, α, is computed as:

$$\alpha = \mathrm{FCDF}^{-1}(1 - P_{FA}, 2, 2k)/k. \tag{C.19}$$

where FCDF represents the CDF of the Fisher distribution. It is important to notice that the scaling factor given by $\frac{N_{\text{Synch}}\sigma^2}{2}$ is the same for Z_{REF} and PDP(.), and consequently, it vanishes when the ratio of the two central Chi-squared random variables is taken.

C.1.3 Step # 3: Detection of the Synchronization Sequence

Finally, after the calculation of the noise-power reference, Z_{REF}, and the synchronization sequence detection threshold level, α, the decision about whether the synchronization sequence is present or not is evaluated through the following hypothesis testing:

$$\mathrm{PDP}(i) \geq \alpha Z_{\text{REF}}, \forall\ i = 0, 1, \cdots, N_{\text{Synch}} - 1. \tag{C.20}$$

If the hypothesis testing is true, then the synchronization sequence plus noise hypothesis, H_1, is selected. Otherwise, if the hypothesis testing is false, then the noise-only hypothesis, H_0, is selected, and consequently, the received signal buffer is proclaimed as not having the presence of the synchronization sequence.

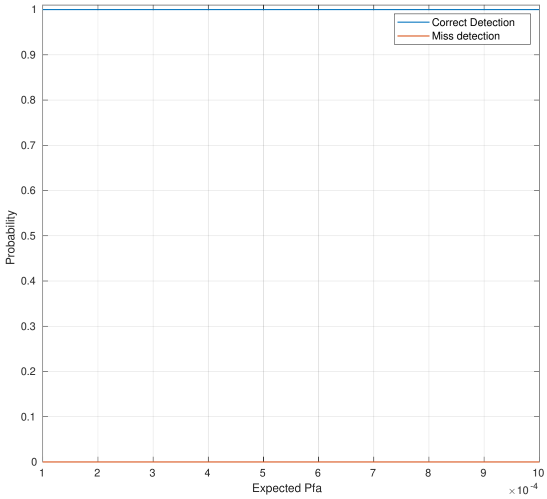


Figure C.4: Comparison between P_{FA} and the actual P_{CD} value for an SNR of 0 *dB.*

C.1.4 Results of the Synchronization Sequence Detection Algorithm

In this section, we present some results when the synchronization sequence detection algorithm is employed. Figure C.3 presents a comparison between the expected P_{FA} and the actual P_{FA} value. The expected P_{FA} was varied from 10^{-5} to 10^{-4}. The number of Monte Carlo trials was set to 10^6 iterations, and an Additive white Gaussian Noise (AWGN) channel was assumed. As can be seen, the actual P_{FA} value stayed very close to the expected one.

Figure C.4 depicts P_{FA} versus the probability of correct detection, P_{CD}. The P_{FA} was varied from 10^{-5} to 10^{-4}; the SNR was set to 0 dB; and the number of Monte Carlo trials was equal to 10^5 iterations. The figure shows that the presence of a synchronization sequence was always detected independent of the P_{FA} value set.

Next, we present some results comparing the detection performance of the proposed two-stage detection algorithm against the single-stage one. The number of Monte Carlo trials here was set to 10^5 iterations, and the AWGN channel was used. For both results shown in Figure C.5, the probability of false alarm P_{FA} was set to 10^{-4}, and the probability of false disposal P_{FD} was set to 10^{-3}. Figure C.5(a) shows the correct and misdetection probabilities for the SNR varying from -12 dB up to 0 dB. As can be noticed, the proposed two-stage detection scheme presented a probability of correct detection of one for SNR values greater than -4 dB, i.e., the presence of the synchronization sequence was always detected, while the single scheme only achieved a probability of one for SNR values greater than 2 dB. Figure C.5(b) shows the correct miss and false detection probabilities when the noise power varied from -30 dBW up to 0 dBW. As can be seen, both

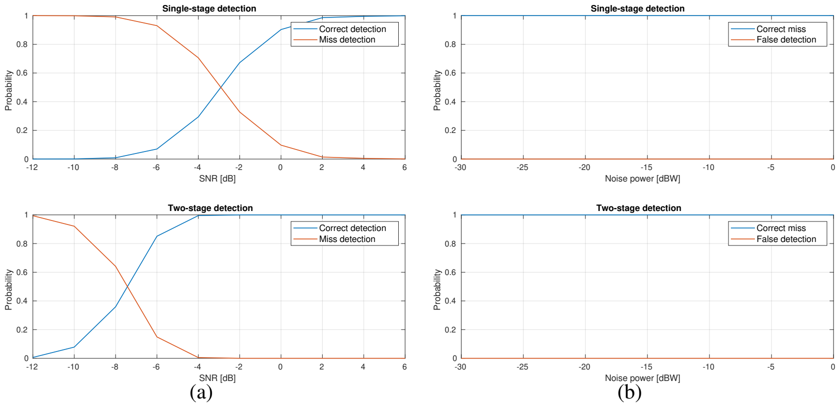


*Figure C.5: Comparison between the single- and two-stage synchronization sequence detection schemes: (**a**) correct and misdetection probabilities; (**b**) correct miss and false detection probabilities.*

schemes presented the correct miss probability equal to one and the false detection probability equal to zero for all noise power values considered in this experiment. It is important to be mentioned that the combined working of the two detection stages decreased the probability of false detection (also known as false alarm) of the two-stage algorithm when compared to the probability of false detection of the second stage alone (see Figure C.3).

References

[1] Sesia, S.; Toufik, I.; Baker, M. *LTE—The UMTS Long Term Evolution: From Theory to Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2011.

[2] Felipe A. P. de Figueiredo, Fabbryccio A. C. M. Cardoso, Jose A. Bianco F., Rafael M. Vilela, Karlo G. Lenzi, *Multi-stage Based Cross-Correlation Peak Detection for LTE Random Access Preambles*, Revista Telecomunicacoes, vol. 15, September 2013.

[3] J. Neyman and E. S. Pearson, *On the problem of the most efficient tests of statistical hypotheses*, Philosophical Transactions of the Royal Society of London, Series A. 231: 289-337, 1933.

[4] Janne J. Lehtomaki, Markku Juntti and Harri Saarnisaari, *CFAR Strategies for Channelized Radiometer*, IEEE Signal Processing Letters, vol. 12, no. 1, January, 2005.

[5] J. G. Proakis, *Digital Communications*, 5th Edition, McGraw-Hill, New York, 2001.

[6] Papoulis, Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th Edition, McGraw-Hill Europe, January 2002.

[7] M. H. DeGroot, *Probability and Statistics*, 2nd Edition, Addison-Wesley, 1986.

D

CP-based CFO Estimation

In this appendix, we propose a CP-based CFO estimation method, which is implemented with SIMD instructions.

D.1 CFO Estimation

During DARPA competition, we noticed our system was struggling to qualify for the final event as we were not meeting the expected throughput. PHY was not decoding high MCS packets. During our testings, we checked channel estimation and time synchronization, which were working fine. Then we decided to check CFO measurements during some matches at the testbed (Colosseum) and noticed CFO was varying a lot during a game. After working on the CFO improvements presented in chapter 5 and here, we qualified for the final event.

D.1.1 Integer Estimation

After Synch signal detection and subframe alignment, the integer (i.e., coarse) Carrier Frequency Offset (CFO) estimation is carried out by correlating the received OFDM symbol carrying the Synch signal with several local versions of it with different integer frequency offset applied to each of them. The PDP between the received and a given offset local version of the Synch OFDM symbol is defined as:

$$\text{PDP}(l, k_{\text{IFO}}) = \left| \sum_{n=0}^{N_{\text{FFT}}-1} y[n] x_l^*[(n+l)_{N_{\text{FFT}}}] e^{\frac{j2\pi k_{\text{IFO}} n}{N_{\text{FFT}}}} \right|^2, \quad \begin{array}{l} 0 \leq l \leq N_{\text{FFT}} - 1 \\ -36 \leq k_{\text{IFO}} \leq 36, \end{array} \tag{D.1}$$

where $x_l(n)$ is the offset local version of the Synch symbol, $y(n)$ is the received Synch OFDM symbol signal, N_{FFT} is the FFT length, $(.)_{N_{\text{FFT}}}$ denotes that this is a cyclic correlation, k_{IFO} is the integer frequency offset, and $(.)^*$ is the complex conjugate. Therefore, the integer frequency offset can be estimated by solving the following maximization problem

$$\hat{\epsilon}_{k_{\text{IFO}}} = \arg \max_{k_{\text{IFO}}} \left(\arg \max_{l} \text{PDP}(l, k_{\text{IFO}}) \right). \tag{D.2}$$

Figure D.1 shows the performance of the integer CFO estimation algorithm described above for an AWGN channel over several SNR values and a PHY BW of 4.5 MHz.

As can be seen, (D.2) refers to a 2 dimensional search, which could be quite complex depending on the number of lags and integer frequency offsets. However, we figured out we could do two things to decrease the complexity. First, the cross-correlation between the received OFDM symbol carrying the synchronization signal and one of its local versions is efficiently performed by frequency domain cross-correlation. Its implementation with FFTs and an IFFT is faster than the time-domain version of the correlation. Second, we realized that with the testbed, Colosseum, we would never have integer CFO values greater than +/- 15 kHz, i.e., one sub-carrier. Therefore, we limited the second search to only three possible values, -15 kHz, 0 Hz, and +15 kHz. In the end, the implementation works fine and is not time-consuming.

D.1.2 Fractional Estimation

As shown by the results in Figure 5.6, the cyclic prefix (CP) based fractional CFO estimation outperformed the Synch based fractional CFO one and therefore was the method adopted by SCATTER PHY. Next, we describe the CP based fractional CFO estimation.

$$\psi(k) = \sum_{n=0}^{N_{\text{CP}}-1} \frac{r_d^*(n + k(N_{\text{FFT}} + N_{\text{CP}})) r_d(n + k(N_{\text{FFT}} + N_{\text{CP}}) + N_{\text{FFT}})}{2\pi}, \tag{D.3}$$

$$\hat{\epsilon}_{\text{FFO}} = \frac{1}{N_{\text{symb}}} \sum_{k=0}^{N_{\text{symb}}-1} \arg\left(\psi(k)\right), \tag{D.4}$$

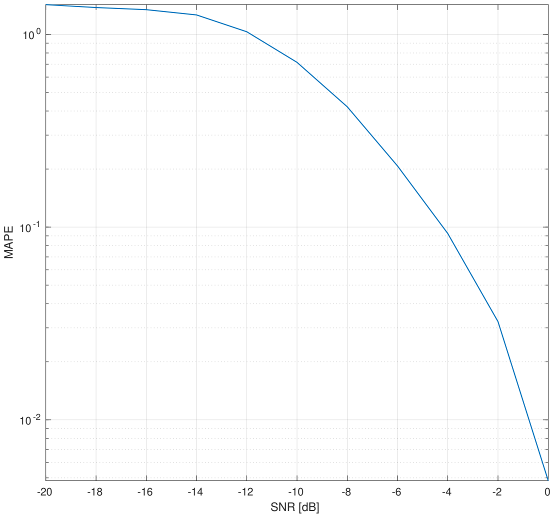
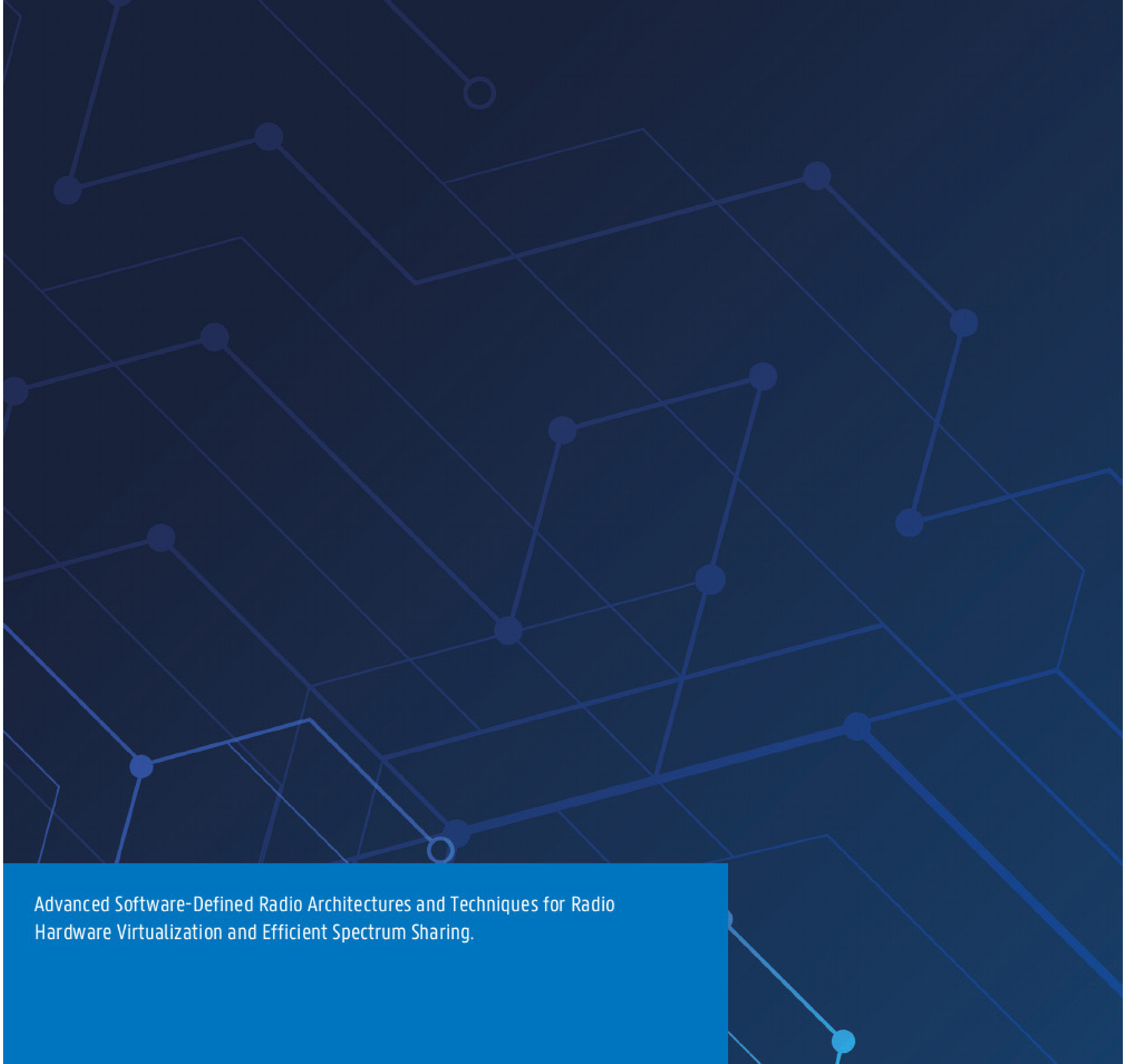


Figure D.1: Performance of the integer CFO estimation algorithm.

where $r_d(n)$ is the detected, integer CFO corrected, and aligned subframe, N_{symb} is the number of OFDM symbols used to average the CFO estimation, and N_{CP} is the length of the OFDM cyclic prefix. As can be seen, the employed CP based fractional CFO estimation was obtained by averaging the estimated fractional CFO over the selected number of OFDM symbols.



Advanced Software-Defined Radio Architectures and Techniques for Radio Hardware Virtualization and Efficient Spectrum Sharing.