Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2017

Tiago Miguel Simões Rosa

Terminal LTE Flexível

Flexible LTE User Equipment

Universidade de Aveiro Departamento de
2017 Eletrónica, Telecomunicações e Informática

Tiago Miguel Simões Rosa

Terminal LTE Flexível

Flexible LTE User Equipment

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Arnaldo Oliveira, professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president	**Professor Doutor Telmo Reis Cunha** Professor Auxiliar da Universidade de Aveiro
vogais / examiners committee	**Professor Doutor Marco Alexandre Cravo Gomes** Professor Auxiliar da Faculdade de Cincias e Tecnologia da Universidade de Coimbra (Arguente)
	Professor Doutor Arnaldo Silva Rodrigues de Oliveira Professor Auxiliar da Universidade de Aveiro (Orientador)

agradecimentos / acknowledgements

Aproveito aqui para deixar um agradecimento a todos os que me apoiaram durante esta fase da minha vida.

Ao meu orientador Professor Doutor Arnaldo Oliveira, pela dedicação, orientação e acompanhamento em cada etapa na realização deste trabalho.

Ao Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro pelas condições de ensino oferecidas, e ao Instituto de Telecomunicações de Aveiro pelas excelsas condições de trabalho proporcionadas durante a elaboração desta dissertação.

Aos meus colegas e amigos, por toda a amizade e apoio demonstrado ao longo destes anos.

Por fim, mas não menos importante, agradeço à minha família, em especial aos meus pais, pelo incondicional apoio, paciência e motivação, sem os quais o cumprimento deste objetivo seria impossível.

A todos, muito obrigado!

palavras-chave LTE, E-UTRA, E-UTRAN, UE, SDR, *Front-end*, OAI, SBC

resumo

As redes móveis estão em constante evolução. A geração atual (4G) de redes celulares de banda larga é representada pelo standard *Long Term Evolution* (LTE), definido pela *3rd Generation Partnership Project* (3GPP). Existe uma elevada procura/uso da rede LTE, com um aumento exponencial do número de dispositivos móveis a requerer uma ligação à *Internet* de alto débito. Isto pode conduzir à sobrelotação do espetro, levando a que o sinal tenha que ser reforçado e a cobertura melhorada em locais específicos, tal como em grandes conferências, festivais e eventos desportivos. Por outro lado, seria uma vantagem importante se os utilizadores pudessem continuar a usar os seus equipamentos e terminais em situações onde o acesso a redes 4G é inexistente, tais como a bordo de um navio, eventos esporádicos em localizações remotas ou em cenários de catástrofe, em que as infraestruturas que permitem as telecomunicações foram danificadas e a cobertura temporária de rede pode ser decisiva em processos de salvamento. Assim sendo, existe uma motivação clara por trás do desenvolvimento de uma infraestrutura celular totalmente reconfigurável e que preencha as características mencionadas anteriormente.

Uma possível abordagem consiste numa plataforma de rádio definido por *software* (SDR), de código aberto, que implementa o standard LTE e corre em processadores de uso geral (GPPs), tornando possível construir uma rede completa investindo somente em *hardware* - computadores e *front-ends* de rádio-frequência (RF). Após comparação e análise de várias plataformas LTE de código aberto foi selecionado o *OpenAirInterface* (OAI) da EURECOM, que disponibiliza uma implementação compatível com a *Release* 8.6 da 3GPP (com parte das funcionalidades da *Release* 10).

O principal objectivo desta dissertação é a implementação de um *User Equipment* (UE) flexível, usando plataformas SDR de código aberto que corram num computador de placa única (SBC) compacto e de baixa potência, integrado com um *front-end* de RF - *Universal Software Radio Peripheral* (USRP). A transmissão de dados em tempo real usando os modos de duplexagem *Time Division Duplex* (TDD) e *Frequency Division Duplex* (FDD) é suportada e a reconfiguração de certos parâmetros é permitida, nomeadamente a frequência portadora, a largura de banda e o número de *Resource Blocks* (RBs) usados. Além disso, é possível partilhar os dados móveis LTE com utilizadores que estejam próximos, semelhante ao que acontece com um *hotspot* de Wi-Fi. O processo de implementação é descrito, incluindo todos os passos necessários para o seu desenvolvimento, englobando o *port* do UE de um computador para um SBC. Finalmente, a performance da rede é analisada, discutindo os valores de débitos obtidos.

keywords LTE, E-UTRA, E-UTRAN, UE, SDR, Front-end, OAI, SBC

abstract

Mobile networks are constantly evolving. 4G is the current generation of broadband cellular network technology and is represented by the Long Term Evolution (LTE) standard, defined by 3rd Generation Partnership Project (3GPP). There's a high demand for LTE at the moment, with the number of mobile devices requiring an high-speed Internet connection increasing exponentially. This may overcrowd the spectrum on the existing deployments and the signal needs to be reinforced and coverage improved in specific sites, such as large conferences, festivals and sport events. On the other hand, it would be an important advantage if users could continue to use their equipment and terminals in situations where cellular networks aren't usually available, such as on board of a cruise ship, sporadic events in remote locations, or in catastrophe scenarios in which the telecommunication infrastructure was damaged and the rapid deployment of a temporary network can save lives. In all of these situations, the availability of flexible and easily deployable cellular base stations and user terminals operating on standard or custom bands would be very desirable. Thus, there is a clear motivation for the development of a fully reconfigurable cellular infrastructure solution that fulfills these requirements.

A possible approach is an open-source, low-cost and low maintenance Software-Defined Radio (SDR) software platform that implements the LTE standard and runs on General Purpose Processors (GPPs), making it possible to build an entire network while only spending money on the hardware itself - computers and Radio-Frequency (RF) front-ends. After comparison and analysis of several open-source LTE SDR platforms, the EURECOM's OpenAirInterface (OAI) was chosen, providing a 3GPP standard-compliant implementation of Release 8.6 (with a subset of Release 10 functionalities).

The main goal of this dissertation is the implementation of a flexible open-source LTE User Equipment (UE) software radio platform on a compact and low-power Single Board Computer (SBC) device, integrated with an RF hardware front-end - Universal Software Radio Peripheral (USRP). It supports real-time Time Division Duplex (TDD) and Frequency Division Duplex (FDD) LTE modes and the reconfiguration of several parameters, namely the carrier frequency, bandwidth and the number of LTE Resource Blocks (RB) used. It can also share its LTE mobile data with nearby users, similarly to a Wi-Fi hotspot. The implementation is described through its several developing steps, including the porting of the UE from a regular computer to a SBC. The performance of the network is then analysed based on measured results of throughput.

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	Third Generation Partnership Program	1
AK	Anonimity Key	90
AKA	Authentication Key and Agreement	4
ALOE++	Abstraction Layer and Open Operating Environment++	30
AP	Access Point	43
APN	Access Point Name	13
ARQ	Automatic Repeat reQuest	15
AS	Access Stratum	11
ASIC	Application-Specific Integrated Circuit	29
AuC	Authentication Centre	13
AUTN	Authentication Token	90
AV	Authentication Vector	90
BER	Bit Error Rate	26
BIOS	Basic Input/Output System	49
BPSK	Binary Phase Shift Keying	26
C-RNTI	Cell Radio Network Temporary Identifier	9
CDMA2000	Code Division Multiple Access 2000	14
CFI	Control Format Indicator	24
CISC	Complex Instruction Set Computers	50
CK	Confidentiality Key	90
CN	Core Network	11
COTS	Commercial off-the-shelf	1
CP	Cyclic Prefix	18
CPU	Central Processing Unit	49
CQI	Channel Quality Indicator	57
CRS	Cell-Specific Reference Signal	25
CSI	Channel State Information	25
CSI-RS	Channel State Information Reference Signal	25

DCI	Downlink Control Information	24
DL	Downlink	6
DL-SCH	Downlink Shared Channel	25
DFT	Discrete Fourier Transform	19
DHCP	Dynamic Host Configuration Protocol	56
DNS	Domain Name System	35
DMRS	DeModulation Reference Signal	25
DSP	Digital Signal Processor	29
DwPTS	Downlink Pilot Time Slot	22
E-UTRA	Evolved Universal Terrestrial Radio Access	6
E-UTRAN	Evolved Universal Terrestrial Radio Access Network	5
EEA	EPS Encryption Algorithm	91
EIA	EPS Integrity Algorithm	91
EIR	Equipment Identity Register	33
eMBMS	Evolved Multimedia Broadcast Multicast Services	32
eMMC	embedded Multi-Media Controller	50
eNB	evolved NodeB	7
EPC	Evolved Packet Core	5
EPS	Evolved Packet System	6
FDD	Frequency Division Duplex	1
FEC	Forward Error Correction	15
FFT	Fast Fourier Transform	21
FIFO	first-in, first-out	51
FPGA	Field-Programmable Gate Array	29
FTP	File Transfer Protocol	7
FQDN	Fully Qualified Domain Name	38
GP	Guard Period	22
GPP	General Purpose Processor	1
GPRS	General Packet Radio System	13
GSM	Global System for Mobile Communications	7
GTP	GPRS Tunelling Protocol	42
GTPv1-U	GPRS Tunelling Protocol-User plane	17
GTPv2-C	GPRS Tunelling Protocol-Control plane	17
GUMMEI	Globally Unique MME Identifier	38
GUTI	Globally Unique Temporary UE Identity	9
HARQ	Hybrid Automatic Repeat reQuest	15

HI	Hybrid-ARQ Indicator	24
HSPA	High Speed Packet Access	13
HSS	Home Subscriber Server	9
I/Q	In-phase/Quadrature	31
IDFT	Inverse Discrete Fourier Transform	20
IETF	Internet Engineering Task Force	16
IFFT	Inverse Fast Fourier Transform	18
IK	Integrity Key	90
IMEI	International Mobile Equipment Identity	9
IMS	IP Multimedia Subsystem	14
IMSI	International Mobile Subscriber Identity	9
IP	Internet Protocol	6
IPv4	IP version 4	9
IPv6	IP version 6	9
IRQ	Interrupt Request	52
ISI	Inter-Symbol Interference	18
ITTI	InTer-Task Interface	36
K	Subscriber Authentication Key	9
L1	Layer 1	14
L2	Layer 2 (data link layer)	14
L3	Layer 3 (network layer)	14
LAN	Local Area Network	56
LTE	Long Term Evolution	1
LTE-R	LTE-Railway	82
LTE-U	LTE-Unlicensed	45
MAC	Medium Access Control	14
MCC	Mobile Country Code	12
ME	Mobile Equipment	7
MIB	Master Information Block	25
MIMO	Multiple Input Multiple Output	34
MME	Mobility Management Entity	10
MMEI	MME Identifier	12
MNC	Mobile Network Code	12
MSISDN	Mobile Station International Subscriber Directory Number	38
MT	Mobile Termination	7
MTU	Maximum Transmission Unit	58

NAS	Non-Access Stratum	11
OAI	OpenAirInterface	1
OFDM	Orthogonal Frequency Division Multiplexing	17
OFDMA	Orthogonal Frequency Division Multiple Access	17
OP	Operator Variant Algorithm Configuration Field	9
OOB	Out-of-Band	26
OSA	OpenAirInterface Software Alliance	33
OSI	Open Systems Interconnection	16
OSLD	Open-Source LTE Deployment	30
P-GW	Packet Data Network Gateway	13
PAPR	Peak-to-Average Power Ratio	18
PBCH	Physical Broadcast Channel	24
PCB	Printed Circuit Board	54
PCEF	Policy Control Enforcement Function	14
PCIe	PCIexpress	52
PCRF	Policy Control and Charging Rules Function	13
PCFICH	Physical Control Format Indicator Channel	24
PDCCH	Physical Downlink Control Channel	24
PDCP	Packet Data Convergence Protocol	14
PDN	Packet Data Network	6
PDSCH	Physical Downlink Shared Channel	24
PHICH	Physical Hybrid-ARQ Indicator Channel	24
PHR	Power Headroom Report	57
PHY	Physical Layer	14
PLMN	Public Land Mobile Network	37
PLMN-ID	Public Land Mobile Network Identity	12
PRACH	Physical Random Access Channel	25
PUCCH	Physical Uplink Control Channel	25
PUSCH	Physical Uplink Shared Channel	24
PRS	Positioning Reference Signal	25
PSS	Primary Synchronization Signal	26
QAM	Quadrature Amplitude Modulation	18
QoS	Quality of Service	6
QPSK	Quadrature Phase Shift Keying	18
RAN	Radio Access Network	11
RB	Resource Block	1

RES	Response	90
RF	Radio Frequency	1
RISC	Reduced Instruction Set Computers	50
RLC	Radio Link Control	14
RNC	Radio Network Controller	33
RRC	Radio Resource Control	14
RS	Reference Signal	25
RTOS	Real-Time Operating System	30
RTT	Round-Trip Time	62
S-GW	Serving Gateway	10
S1AP	S1 Application Protocol	17
SAE	System Architecture Evolution	6
SAP	Service Access Point	16
SAW	Surface Acoustic Wave	4
SBC	Single-Board Computer	2
SC-FDMA	Single-Carrier Frequency Division Multiple Access	17
SCTP	Stream Control Transmission Protocol	17
SDR	Software-Defined Radio	2
SGSN	Serving GPRS Support Node	33
SIM	Subscriber Identity Module	7
SIMD	Single Instruction, Multiple Data	38
SISO	Single Input Single Output	34
SMA	SubMiniature version A	53
SMT	Surface-Mount Technology	54
SNAT	Source Network Address Translation	38
SNR	Signal-to-Noise Ratio	26
SQN	Sequence Number	9
SRB	Special Radio Bearer	37
SRS	Sounding Reference Signal	25
SSH	Secure Shell	52
SSS	Secondary Synchronization Signal	26
TAC	Tracking Area Code	12
TAI	Tracking Area Identity	12
TCP	Transmission Control Protocol	17
TDD	Time Division Duplex	1
TE	Terminal Equipment	7

TETRA	Terrestrial Trunked Radio	45
VoIP	Voice over IP	6
UCI	Uplink Control Information	25
UDP	User Datagram Protocol	17
UE	User Equipment	1
UHD	USRP Hardware Driver	53
UICC	Universal Integrated Circuit Card	7
UL	Uplink	6
UMTS	Universal Mobile Telecommunications System	6
UpPTS	Uplink Pilot Time Slot	22
USB	Universal Serial Bus	52
USIM	Universal Subscriber Identity Module	7
USRP	Universal Software Radio Peripheral	3
VHDL	Very High Speed Integrated Circuit Hardware Description Language	29
WiMAX	Worldwide Interoperability for Microwave Access	14
XRES	Expected Response	90
X2AP	X2 Application Protocol	17

Chapter 1

Introduction

In this chapter, the work developed along this dissertation is introduced. The framework is explained, followed by the reasons that motivated its elaboration. The objectives proposed for the dissertation work are detailed, along with the contributions that were made. To finalize, an outline specifying the contents of this work is presented.

1.1 Framework

The mobile telecommunication technologies are growing at a very fast pace, as well as the number of equipments requiring connection to the network. Compared with the previous generation networks, 4G provides faster transmission rates, bigger network capacity and coverage, lower latency and more flexible bandwidths. The current generation of broadband cellular network technology is represented by the Long Term Evolution (LTE) standard, defined by the Third Generation Partnership Program (3GPP) consortium.

Currently, the hardware/software for mobile networks consists of a large number of closed-source/proprietary elements, making the deployment of new services and technologies costly to the operators and, therefore, restraining innovation. The commercially available terminals, also known as Commercial off-the-shelf (COTS) User Equipments (UEs), are sold to the user with previously set hardware capabilities that can't be altered, which makes them tied to a certain number of frequency bands or a specific duplex mode.

On the other hand, there is open-source software that runs on General Purpose Processors (GPPs) (Intel's x86 or ARM architectures) and implements the LTE stack in real-time, with the benefit of only requiring computers and Radio Frequency (RF) front-ends to run. This software can reduce the cost of implementation, increase the flexibility and simplify the network access, bringing a significant efficiency to the network design [Ope17a]. It is essential for research and experimentation, as it provides the required base to implement and evaluate the performance of new technologies.

There are already a vast number of open-source platforms that implement a LTE network. A comparison between them is provided later, in Chapter 3, and the one that provides the best and most accurate implementation is OpenAirInterface (OAI). It can implement the core network, the access network and, most importantly in the scope of this work, a software radio UE. This UE is reconfigurable and very flexible: it works in any licensed or custom frequency band, in different LTE duplex modes - Frequency Division Duplex (FDD) or Time Division Duplex (TDD) -, with various sizes of LTE Resource Blocks (RBs), or even custom power configurations. Such a flexible and open-source solution can prove to be useful on multi-operational environments, with different contexts of utilization.

1.2 Motivation

There's an high demand for LTE at the moment, with the number of devices requiring an high-speed Internet connection increasing exponentially. Despite of the current 4G LTE network deployments and coverage being globally widespread, there are still scenarios where the availability of flexible and easily deployable cellular base stations and user terminals operating on standard/custom frequency bands would be very desirable. Some of these scenarios include:

- Big events, where the existing deployments cannot handle the amount of devices connected to the network and become overloaded. Thus, the signal must be reinforced and/or the coverage improved. Some examples are: large conferences, festivals and sport events.

- Situations where network deployments aren't usually available and a 4G connection can't be obtained, such as on board of a cruise ship, or for sporadic events in remote locations.

- Places where the mobile networks stop being available, mainly in catastrophe scenarios, such as natural disasters or war situations, in which the telecommunication infrastructure is damaged. The rapid deployment of a temporary network could save lives.

Therefore, there is a clear motivation for the development of a low-cost and fully reconfigurable cellular infrastructure solution that fulfils these requirements.

1.3 Objectives

The main goal of this dissertation is the implementation of a flexible open-source LTE UE software radio platform on a compact and low-power device, integrated with an RF hardware front-end. It must support the LTE duplex modes in real-time - TDD and FDD -, as well as the reconfiguration of several parameters, namely the carrier frequency, bandwidth and custom power configurations. The UE should be able to share its LTE mobile data with nearby users, similarly to a Wi-Fi hotspot.

In an initial phase, some of the existent open-source LTE Software-Defined Radio (SDR) platforms and RF front-ends must be compared, so a decision on the most adequate testing platform and radio hardware can be made. Using the chosen platform, a basic LTE network should then be deployed, interfacing and testing it with different UE implementations - commercial or open-source software radio.

An open-source software radio UE is usually run with the aid of a computer. So, as a second phase and to achieve the compactability required for a flexible and easily deployable device, the computer running the UE should be switched for a smaller platform, such as a Single-Board Computer (SBC).

For testing in different LTE frequency bands, additional RF components must be developed, if required (e.g. duplexers).

As a final objective, a performance analysis of the system must be made, mainly based on measured results of throughput, followed by a discussion of the results.

1.4 Contributions

The main contributions of this dissertation are:

- The implementation of a flexible and compact open-source software radio LTE UE on an Intel-based SBC (*UP Squared*), using OAI's software, interfaced with an Universal Software Radio Peripheral (USRP) as RF front-end. It can be configured to transmit in any licensed, unlicensed and custom frequency band, as well as with both LTE duplex modes (FDD and TDD). Its LTE mobile data connection can be shared with the surrounding devices by the means of a Wi-Fi hotspot.

- The performance analysis of the implementation, based mainly on measured results of throughput and latency.

1.5 Outline

In addition to this introductory chapter, this document is divided into six chapters with the following contents:

- **Chapter 2 - "Long Term Evolution (LTE)":** provides a background on the LTE standard, required for the development and understanding of this dissertation. The network architecture is described, detailing each of the relevant nodes and identifiers. The protocol stack is presented, followed by a deeper description of the physical layer, where the multiple access technologies, frame structure of the duplex modes and physical channels and signals are the main focus. The modulation schemes used in LTE are then presented, finishing with a selection of the many existing LTE frequency bands.

- **Chapter 3 - "LTE Experimentation Frameworks":** presents several of the existing experimentation frameworks that implement the LTE standard and its main network elements: the user equipment, access network and core network. After a brief comparison between the frameworks, the one providing the best trade-off is selected as a point of departure for the implementation of the proposed dissertation work and subsequently described.

- **Chapter 4 - "Non-Conventional LTE Deployment Scenarios":** describes the deployment possibilities when using OAI's framework, followed by a listing of non-conventional LTE deployment scenarios that would benefit from a flexible and compact LTE UE implementation.

- **Chapter 5 - "Implementation":** describes the implementation of the setup, starting with a block diagram of the network components and their radio configurations, which are then detailed in different sections. Information on how the obtained LTE mobile data is shared by creating a free Wi-Fi hotspot is also given.

- **Chapter 6 - "Results":** provides a description of the implemented setup and the tools used for testing. Three UE implementations with different radio configurations were analysed, one using commercial UEs and the other two using software-defined radio UEs running on a mini-computer and single-board computer, respectively. The measured throughput and latency results were presented, along with additional relevant parameters.

- **Chapter 7 - "Conclusions and Future Work":** concludes the dissertation work and introduces future work possibilities.

In order to provide additional information to the dissertation, the following appendixes were included:

- **Appendix A - "LTE Authentication Procedure":** provides a detailed description of the LTE mutual authentication procedure, including the Authentication Key and Agreement (AKA) process and integrity and ciphering algorithms.
- **Appendix B - "Passive Surface Acoustic Wave (SAW) Duplexers Design":** gives an insight on the board design and subsequent S-parameters measurement of the developed passive SAW duplexers, for LTE frequency bands 1, 5 and 7.
- **Appendix C - "OpenAirInterface Installation and User Guide":** developed in the context of this dissertation, gathers all the installation steps required for installing, configuring and executing the OAI platform in its current development status. It provides the configuration files used for the several network components, along with a description on how to add custom/unlicensed frequency bands to the system, as well as the steps required for sharing the LTE network through a Wi-Fi hotspot.
- **Appendix D - "3D Printed Flexible UE Prototype Holder":** shows the development process of a 3D printed holder designed to turn the flexible UE implementation into a more compact solution. It holds the single-board computer *UP Squared* and the RF front-end *USRP B200mini*.

Chapter 2

Long Term Evolution (LTE)

This chapter provides a background on LTE, required for the development and understanding of this dissertation. It begins with a description of a basic network architecture, followed by an explanation of LTE's network architecture in section 2.2, divided in UE, Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC) and detailing each of their relevant nodes and identifiers. The LTE protocol stack (along with its relevant interface protocols) is then presented in section 2.3, followed by a deeper description of the physical layer, in section 2.4, where the multiple access technologies, frame structure of the duplex modes and physical channels and signals are the main focus. The modulation schemes used in LTE are then presented in section 2.5, finishing with section 2.6, which displays a selection of the many existing LTE frequency bands.

2.1 Introduction

A basic mobile telecommunications network is divided into three main parts: the user equipment, the access network and the core network. The core network is the backbone of a network, providing its subscribers with services like Internet access, voice call support or access to other networks. It is where all the information regarding the subscribers is stored, mainly used for the authentication process, where the network determines if the subscriber requesting a service is authorized to do so. In Figure 2.1, where a basic mobile network architecture is depicted, it is possible to see the core network being connected with one of the services it provides, Internet access. Usually, a core network is located in some sort of building, and it is supposed to be stationary. In order to connect the subscribers to their service provider, an access network is required.

The access network is the “outside” network that allows the subscribers to connect to the core network, wirelessly. It is composed by a great number of radio equipment, such as cables and antennas, which are usually mounted on base station towers or tall buildings spread around a certain area. The access network is directly connected to the core network, as represented in Figure 2.1.

The final network component is the user equipment, a terminal device used by the subscriber for communication. It connects to the access network through a wireless radio interface, which then redirects the exchanged information to/from the core network. The user equipment could be any device that uses mobile broadband communications, such as a smartphone or a computer with a mobile broadband adapter (commonly named dongle).

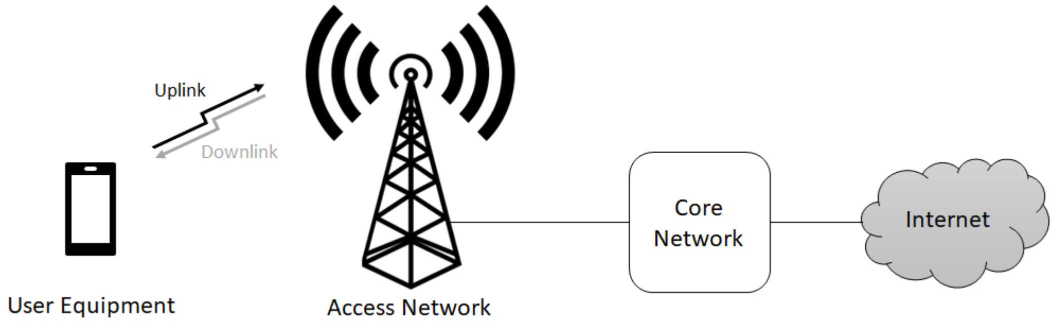


Figure 2.1: Architecture of a basic cellular telecommunications network.

In Figure 2.1, it is possible to notice that the radio interface between the user equipment and the access network was divided in two links, the Uplink (UL) and Downlink (DL), with arrows pointing in opposite directions. This terminology is used to distinguish the communications from the user equipment to the access network (uplink), and the communications from the access network to the user equipment (downlink).

2.2 Network Architecture

After the worldwide deployment of 3G Universal Mobile Telecommunications System (UMTS) systems, there was a need to ensure they would remain competitive in the future. In November 2004, 3GPP began a project to define the long-term evolution of UMTS cellular technology. The specifications related to this effort are formally known as Evolved Universal Terrestrial Radio Access (E-UTRA) and E-UTRAN, describing the **user equipment** and the **access network**, respectively, but are usually referred to by the project name LTE - Long Term Evolution. This fourth-generation mobile telecommunications network began to be documented in Release 8 of the 3GPP specifications. 3GPP's requirements for LTE include reduced cost per bit, better service provisioning, flexible use of new and existing frequency bands, a simplified network architecture with open interfaces and an allowance for reasonable power consumption by terminals.

A parallel 3GPP project, called System Architecture Evolution (SAE), defines an all-Internet Protocol (IP), packet-only **core network** that aims to deliver the higher throughput, lower cost and lower latency promised by LTE. Its main component is the EPC. The EPC is also designed to provide seamless interworking with existing 3GPP and non-3GPP access technologies [Mor08, Agi09].

The combination of the EPC with the evolved access network - E-UTRAN - and the user equipment - E-UTRA - comprises the Evolved Packet System (EPS), whose architecture is depicted in Figure 2.2.

EPS uses the concept of *EPS bearers* to route IP traffic from a gateway in the Packet Data Network (PDN) to the UE, providing Internet access, as well as services like Voice over IP (VoIP) to the user. A bearer is an IP packet flow with a defined Quality of Service (QoS). The E-UTRAN and EPC set up and release bearers as required by applications. Multiple bearers can be established for a user in order to provide different QoS streams or connectivity to different PDNs. For example, an user might be engaged in a VoIP call while at the

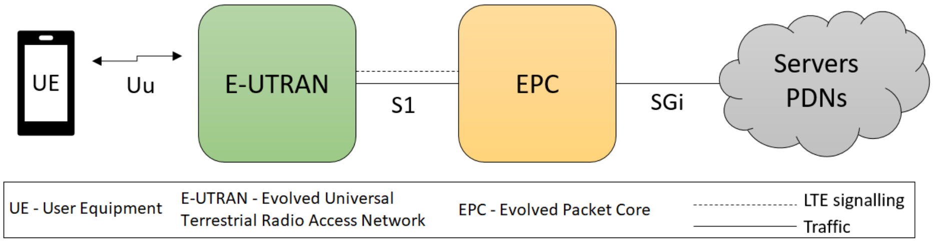

UE
Uu
E-UTRAN
S1
EPC
SGi
Servers
PDNs
UE - User Equipment
E-UTRAN - Evolved Universal Terrestrial Radio Access Network
EPC - Evolved Packet Core
LTE signalling
Traffic

Figure 2.2: Evolved Packet System (EPS) network architecture.

same time performing web browsing or a File Transfer Protocol (FTP) download. A VoIP bearer would provide the necessary QoS for the voice call, while a best-effort bearer would be suitable for web browsing or FTP session. The network must also provide sufficient security and privacy for the user and protection for the network against fraudulent use.

All of this is achieved by means of the EPS network elements (simplified in Figure 2.2), which are interconnected by standardized interfaces in order to allow for interoperability with multiple vendors. This gives network operators the possibility to source different network elements from different vendors, as well as splitting or merging them depending on commercial considerations. While the core network (EPC) consists of many logical nodes, the access network (E-UTRAN) is made up of essentially just one, the evolved NodeB (eNB), which connects to the UEs [Ste11, Alc09]. These network elements are described in more detail in the following sections and are depicted in Figure 2.4.

2.2.1 User Equipment

Figure 2.3 shows the internal architecture of the User Equipment (UE). The architecture is identical to the previous generations UMTS and Global System for Mobile Communications (GSM). The actual communication device is known as the Mobile Equipment (ME), in case of it being a voice mobile or a smartphone. However, the mobile equipment can be divided into two components, namely the Mobile Termination (MT), which handles all the communication functions, and the Terminal Equipment (TE), which terminates the data streams. The mobile termination might be a plug-in LTE card for a laptop, for example, meaning the terminal equipment would be the laptop itself.

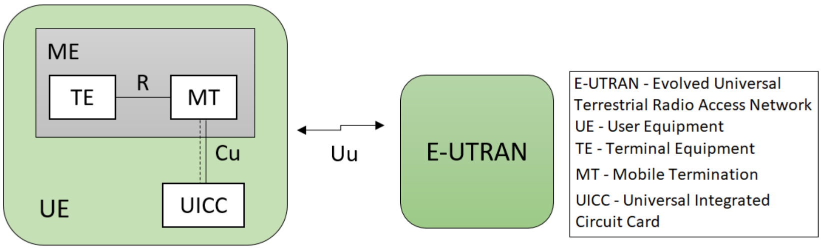

ME
TE
R
MT
Cu
UE
UICC
Uu
E-UTRAN
E-UTRAN - Evolved Universal Terrestrial Radio Access Network
UE - User Equipment
TE - Terminal Equipment
MT - Mobile Termination
UICC - Universal Integrated Circuit Card

Figure 2.3: UE's architecture (adapted from [Chr12]).

The Universal Integrated Circuit Card (UICC) is a smart card, colloquially known as the Subscriber Identity Module (SIM) card. It runs an application known as the Universal Subscriber Identity Module (USIM), which stores user-specific data such as the user's phone

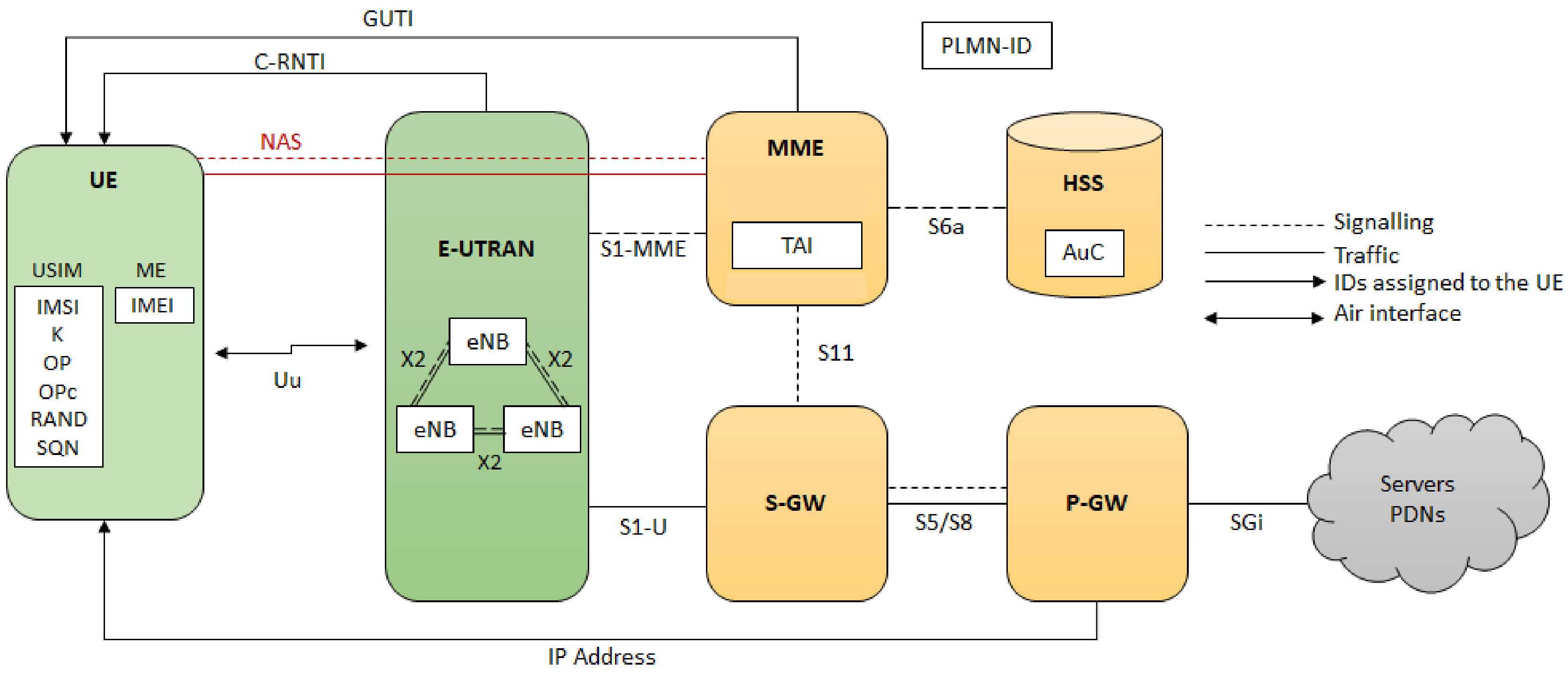


Figure 2.4: Overall LTE network architecture.

number and home network identity. The USIM carries out various security-related calculations and algorithms, using secure keys that the smart card stores. LTE supports UEs that are using a USIM from Release 99 or later, but it does not support the SIM that was used by earlier releases of GSM.

In addition, LTE supports UEs that are using IP version 4 (IPv4), IP version 6 (IPv6), or dual stack IPv4/IPv6. An UE receives one IP address for every packet data network that it is communicating with, e.g. one for the Internet and one for any private corporate network. Alternatively, the UE can receive an IPv4 address together with an IPv6 address, if the two versions of the protocol are supported.

UEs can have a wide variety of radio capabilities, which cover issues such as the maximum data rate they can handle, the different types of radio access technology they support and the carrier frequencies on which they can transmit and receive. These capabilities are passed to the radio access network by means of signalling messages, so that the E-UTRAN knows how to control them correctly [Chr12]. The most important capabilities are grouped together into the *UE category*, shown in Tables 4.1-1 and 4.1-2 of [3GP16b].

In sections 2.2.1.1 and 2.2.1.2, some important parameters regarding the UE are introduced. First, UE's unique identities are discussed, followed by parameters regarding the security of the UE.

2.2.1.1 UE Identifiers

The International Mobile Subscriber Identity (IMSI) and International Mobile Equipment Identity (IMEI) are unique identities assigned to the USIM card and the ME, respectively. They are permanently associated with the subscriber and stored in a provider database like the Home Subscriber Server (HSS) (described in section 2.2.3.2) and are used by other nodes in the network to identify the user equipment. Similar to 2G and 3G technologies, due to security and efficiency reasons, the LTE network minimizes the exchange of these two identifiers with the UE.

During the initial attach procedure between the UE and the LTE network, the UE is assigned three additional dynamic identifiers: a Cell Radio Network Temporary Identifier (C-RNTI), discussed in section 2.2.2.1; a Globally Unique Temporary UE Identity (GUTI), described in section 2.2.3.1; an IP address, discussed in section 2.2.3.4 [3GP14a].

2.2.1.2 Security and USIM Card Parameters

The security architecture of the 3GPP system is composed of several security functions. These functions are not standardized, so each network operator can design their own. The USIM card is programmed with several parameters (also stored in the network's database, HSS, described in section 2.2.3.2), which are used as inputs to these security functions and algorithms. The USIM card parameters are used to authenticate the UE to the network, by comparing them with the ones stored in the network's database. Most of these parameters are never shared in the network, due to security reasons, and the most important are: Subscriber Authentication Key (K), Operator Variant Algorithm Configuration Field (OP), Sequence Number (SQN) (incremented in each authentication) and a randomly generated User Authentication Challenge (RAND).

Each network operator will define a value for OP, which will then be used for all of its subscribers. As cracking this value could lead to a potential spoof of all subscriber's USIM cards, an additional essential parameter is required: OPc. The OPc is a key derived from the OP and K, by using an encryption algorithm, and will substitute the OP in subsequent computations, so it can be kept secret. It is impossible to obtain the OPc from the OP, using reverse engineering.

2.2.2 Access Network

Figure 2.5 illustrates LTE's access network. The Evolved Universal Terrestrial Radio Access Network (E-UTRAN) consists of a network of evolved NodeBs (eNBs), providing the E-UTRA user plane and control plane protocol terminations towards the UE (more information regarding protocols are provided on section 2.3).

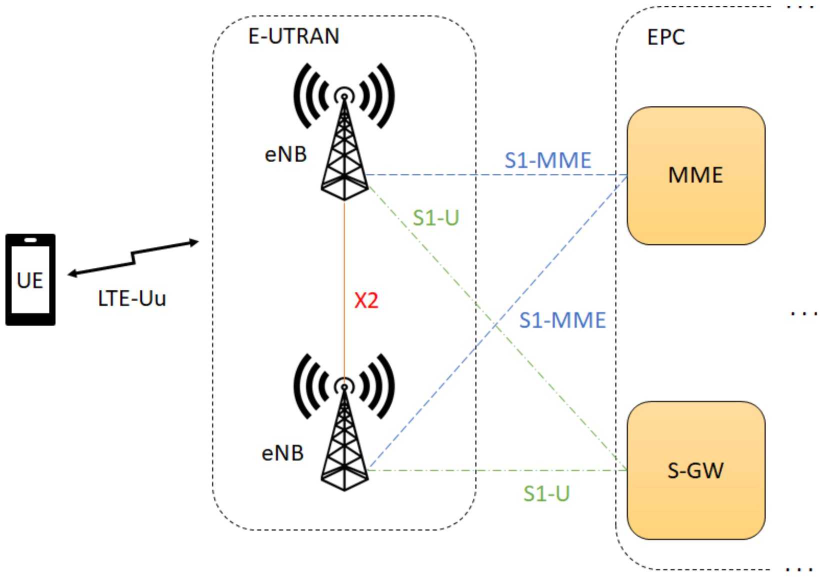


Figure 2.5: E-UTRAN's architecture.

The eNBs are interconnected by means of the X2 interface, used for signalling and to support lossless mobility between neighbouring cells through packet forwarding. If the E-UTRAN is only composed of a single eNB, no X2 interface is required.

The eNBs are also connected by means of the S1 interface to the core network (EPC, described in section 2.2.3), more specifically to the Mobility Management Entity (MME) with the S1-MME interface and to the Serving Gateway (S-GW) with the S1-U interface [3GP15b]. The S1-MME is used for control plane protocols (described later in section 2.3) and the S1-U for user plane data exchange with the core network's serving gateway (presented in section 2.2.3.3).

The interface between the E-UTRAN and the UE is known as the LTE-Uu interface. Unless dual connectivity is used, an UE is connected to a single eNB at a time [Eri16].

It is important to note that an eNB is a logical node and not a physical implementation. One common implementation of an eNB is a three-sector site, where a base station is handling transmissions in three cells, although other implementations can be found as well, such as one baseband processing unit to which a number of remote radio heads are connected. One example of the latter is a large number of indoor cells, or several cells along a highway, belonging to the same eNB. Thus, a base station is a possible implementation of, but not the same as, an eNB.

The E-UTRAN does not have a centralized controller, as in previous generations of mobile networks, hence its architecture is said to be flat. Instead, each eNB is responsible for all radio-related functions in one or several cells, which can be summarized as:

- **Radio Resource Management:** covers all functions related to the radio bearers, such as radio bearer control, radio admission control, radio mobility control, scheduling and dynamic allocation of resources to UEs in both uplink and downlink.

- **Header Compression:** helps to ensure an efficient use of the radio interface by compressing the IP packet headers which could otherwise represent a significant overhead, especially for small packets such as VoIP.

- **Security:** encrypts all data sent over the radio interface.

- **Connectivity to the EPC:** ensures the signalling towards the MME and the bearer path towards the S-GW are maintained.

Not having a centralized controller allows for a tighter interaction between the different protocol layers of the radio access network (described in section 2.3), thus reducing latency and improving efficiency [Ste11].

The E-UTRAN assigns an unique identifier to an UE, named C-RNTI, and it is described in the following section. This is how the access network identifies the UE when it is attached to the network.

2.2.2.1 Cell Radio Network Temporary Identifier (C-RNTI)

The eNB assigns the UE a Cell Radio Network Temporary Identifier (C-RNTI) to identify it during the exchange of information over the air. This entity is assigned during the UE attach procedure and is only valid for as long as the UE and eNB are connected. Once the UE leaves the coverage area of an eNB, a new C-RNTI must be assigned. The C-RNTI is an E-UTRAN specific identifier and the EPC has no visibility to it.

2.2.3 Core Network

The core network, also known as Evolved Packet Core (EPC), is responsible for the overall control of the UE and the establishment of bearers. The main logical nodes of the EPC are shown in Figure 2.6 and discussed in more detail below, as well as some additional functions and entities.

2.2.3.1 Mobility Management Entity (MME)

The MME is the control-plane node which processes the signalling messages between the UE and the EPC unrelated to radio communications. The protocols running between the UE and the Core Network (CN) are referred to as the Non-Access Stratum (NAS) protocols, to separate it from the Access Stratum (AS) which handles the protocols operating between the UE and the Radio Access Network (RAN). These protocols will be described in section 2.3. The main functions supported by the MME are classified as:

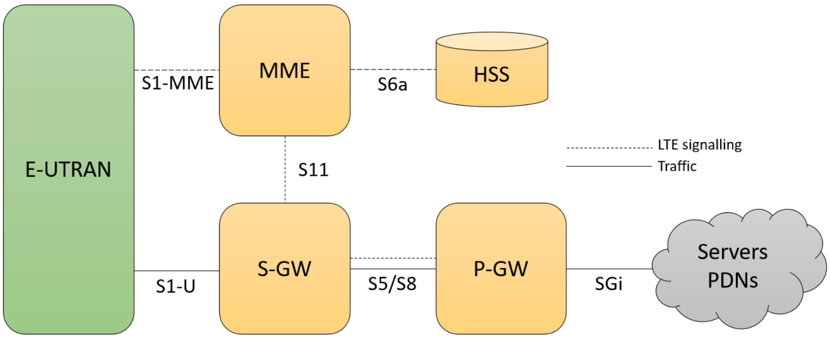


Figure 2.6: EPC's architecture (adapted from [Eri16, Chr12]).

- Functions related to bearer management: responsible for the establishment, maintenance and release of the bearers, and are handled by the session management layer in the NAS protocol.
- Functions related to connection management: responsible for the establishment of the connection and security between the network and UE, and are handled by the connection or mobility management layer in the NAS protocol.
- Functions related to interworking with other networks: responsible for the handing over of voice calls to legacy networks.

A typical network might contain a handful of MMEs, each of which looks after a certain geographical region. Each device is assigned to a single MME, which is known as its serving MME, but that can be changed if it moves sufficiently far.

An MME is identified globally by merging an MME Identifier (MMEI), which identifies an MME inside a particular network, with the network identifier, Public Land Mobile Network Identity (PLMN-ID). Another identifier is the Tracking Area Identity (TAI), where the MME defines its tracking area. When an UE connects to the network, the MME will assign it a temporary identity, named GUTI. All of these identifiers are detailed below.

Public Land Mobile Network Identity (PLMN-ID)

As in previous generations, each network is associated with a Public Land Mobile Network Identity (PLMN-ID). It comprises a Mobile Country Code (MCC) and a Mobile Network Code (MNC). The MCC for Portugal is 268, while the different network operators are defined by the MNC: 01 for Vodafone, 03 for NOS and 06 for altice MEO. Consequently, the PLMN-ID is 26801, 26803 and 26806 for the Portuguese network operators, respectively.

Tracking Area Identity (TAI)

A tracking area is a predefined location where an UE can move freely, without updating the MME. Tracking areas are used to track the locations of UEs that are on idle mode, as in active state their location is known by the LTE network. Each tracking area has two main identities. The Tracking Area Code (TAC) identifies a tracking area within a particular network, which when combined with the network identity (or PLMN-ID) forms the globally unique Tracking Area Identity (TAI).

Globally Unique Temporary UE Identity (GUTI)

In order to protect the user's permanent identity in the EPS, the serving MME identifies an UE using temporary identities, which are updated at regular intervals [Chr12]. During the attach procedure, the MME assigns a GUTI to the UE, used for identification purposes during all message exchanges and procedures with the EPC. It is only valid for as long as the UE remains attached to the MME. A new GUTI will be assigned once the UE leaves the MME's tracking area. Embedded within the GUTI are the PLMN-ID of the service provider and the MMEI, as previously mentioned [Ste11, Eri16, 3GP14b, 3GP14a, Awa10].

2.2.3.2 Home Subscriber Server (HSS)

The HSS is a central database that contains users' SAE subscription data such as the EPS-subscribed QoS profile and any access restrictions for roaming. It also holds information about the PDNs to which the user can connect. This could be in the form of an Access Point Name (APN) (which is a label describing the access point to the PDN), or a PDN Address (indicating subscribed IP addresses). In addition, the HSS holds dynamic information such as the identity of the MME to which the user is currently attached or registered. The S6a interface it uses to communicate with the MME is described in section 2.3.1. The HSS may also integrate an Authentication Centre (AuC).

Authentication Center (AuC)

The AuC generates the vectors for authentication and security keys. It is one of the few components of LTE that has been carried forward from UMTS and GSM [Chr12, Eri16].

2.2.3.3 Serving Gateway (S-GW)

The S-GW is the user-plane's node which forwards all user IP packets between the E-UTRAN and the Packet Data Network Gateway (P-GW) (described in the following section). The S-GW acts as a mobility anchor when devices move between eNBs, as well as a mobility anchor for other previous 3GPP technologies - GSM/General Packet Radio System (GPRS) and High Speed Packet Access (HSPA). It retains information about the bearers when the UE is in idle state and temporarily buffers downlink data while the MME initiates paging of the UE to re-establish the bearers. In addition, the Serving Gateway (S-GW) performs some administrative functions in the network, such as collecting information and statistics necessary for charging (e.g. the volume of data sent to or received from the user). As with the MME, a typical network might contain a handful of serving gateways, each of which looks after the mobiles in a certain geographical region. Each mobile is assigned to a single S-GW, but it can be changed if the mobile moves sufficiently far [Chr12].

2.2.3.4 Packet Data Network Gateway (P-GW)

The Packet Data Network Gateway (P-GW) is the EPC's point of contact with the outside world. It is responsible for IP address allocation for the UE, as well as QoS enforcement and flow-based charging according to rules from the Policy Control and Charging Rules Function (PCRF), described in section 2.2.3.5. Through the SGi interface, each P-GW exchanges data with one or more external devices or packet data networks, such as the network operator's servers, the Internet or the IP Multimedia Subsystem (IMS). Each packet data network is identified by an APN. A network operator typically uses a handful of different APNs (e.g., one for its own servers and one for the internet). Each UE is assigned to a default P-GW when it first switches on, to give it always-on connectivity to a default PDN such as the Internet. Later on, a UE may be assigned to one or more additional PDN gateways, if it wishes to connect to additional packet data networks, such as private corporate networks.

The P-GW remains the same throughout the lifetime of the data connection. It also serves as the mobility anchor for interworking with non-3GPP technologies such as Code Division Multiple Access 2000 (CDMA2000) and Worldwide Interoperability for Microwave Access (WiMAX) networks [Chr12, Eri16]. The P-GW assigns the UE with an IP address when it connects to the network, as described below.

IP Address

The P-GW assigns the UE an IP address (IPv4 and/or IPv6) to facilitate data connectivity between the UE and any internal or external PDN. The UE may be connected to more than one P-GW, so it may be assigned more than one IP address. The first IP address is assigned to the UE during the Initial Attach procedure and it stays with the UE as long as it is attached to the network. Unlike the other temporary identifiers, the IP address does not change while the UE is attached. It is used for IP forwarding decisions and the eNB, MME and S-GW have no use for it [Awa10].

2.2.3.5 Policy Control and Charging Rules Function(PCRF)

The PCRF is responsible for policy control decision-making, as well as for controlling the flow-based charging functionalities in the Policy Control Enforcement Function (PCEF), which resides in the P-GW. The PCRF provides the QoS authorization that decides how a certain data flow will be treated in the PCEF and ensures that this is in accordance with the users subscription profile [Ste11].

The previously discussed nodes are logical nodes. In an actual physical implementation, several of them may be combined into a single physical node, for example, the P-GW and S-GW [Eri16].

2.3 Protocol Stack

The functional split between the EPC and E-UTRAN is shown in Figure 2.7. It represents the layered protocol stacks, which can be divided into user plane and control plane. The user plane is composed by the sub-layers Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC), which form Layer 2 (data link layer) (L2), and the Physical Layer (PHY), also known as Layer 1 (L1). The control plane additionally includes the Radio Resource Control (RRC), commonly named Layer 3 (network layer) (L3). Together, these layers form a protocol stack known as the Access Stratum (AS). The upper layer in the control plane, which terminates in the UE and in the MME, is referred to as Non-Access Stratum (NAS), whose protocols are completely independent of the access technology.

These protocols and their main functionalities are summarized below [Ste11, 3GP14c, 3GP14d, 3GP11b, 3GP17b, 3GP11a, Eri16]:

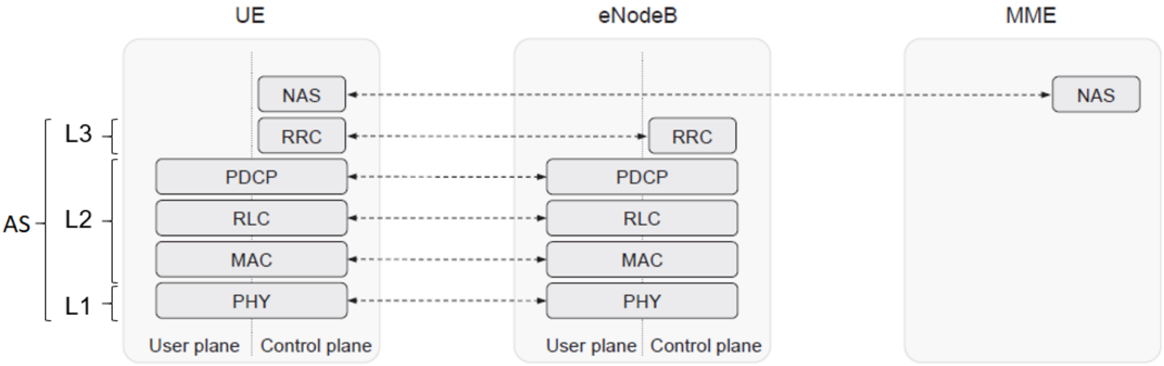


Figure 2.7: Functional split between the E-UTRAN and EPC (adapted from [Eri16]).

- **Non-Access Stratum (NAS)**: the NAS protocol is handled by the MME and includes features such as EPS bearer management, authentication, security control procedures and different idle-mode procedures (e.g. paging). It is also responsible for assigning an IP address to a device.

- **Radio Resource Control (RRC)**: the RRC layer is responsible for handling the RAN-related procedures, including broadcast of system information necessary for the device to be able to communicate with a cell, transmission of paging messages originating from the MME (to notify the device about incoming connection requests), connection management (including the establishment of radio bearers), mobility functions such as cell (re)selection, measurement configuration and reporting, and handling of UE capabilities. It is also responsible for configuring all the lower layers.

- **Packet Data Convergence Protocol (PDCP)**: the PDCP layer processes RRC messages in the control plane and IP packets in the user plane. The main functions of the PDCP layer are IP header compression, security (integrity protection and ciphering) and support for reordering and retransmission during handover.

- **Radio Link Control (RLC)**: the RLC layer performs segmentation and reassembly of upper layer packets, in order to adapt them to the size which can actually be transmitted over the radio interface, as well as error correction through Automatic Repeat reQuest (ARQ).

- **Medium Access Control (MAC)**: the MAC layer, as illustrated in Figure 2.8, handles the mapping between logical channels (upper layer) and transport channels (lower layer). It also handles error correction through Hybrid Automatic Repeat reQuest (HARQ) and uplink and downlink scheduling information reporting.

- **Physical Layer (PHY)**: the physical layer provides data transport services to the MAC layer through the use of transport channels. It also handles time/frequency resource mapping, Forward Error Correction (FEC) coding/decoding, modulation/demodulation of physical channels, multi-antenna processing, amongst others.

To efficiently support various classes of services, LTE adopts a hierarchical channel structure. As previously mentioned, there are three different channel types defined in LTE, each

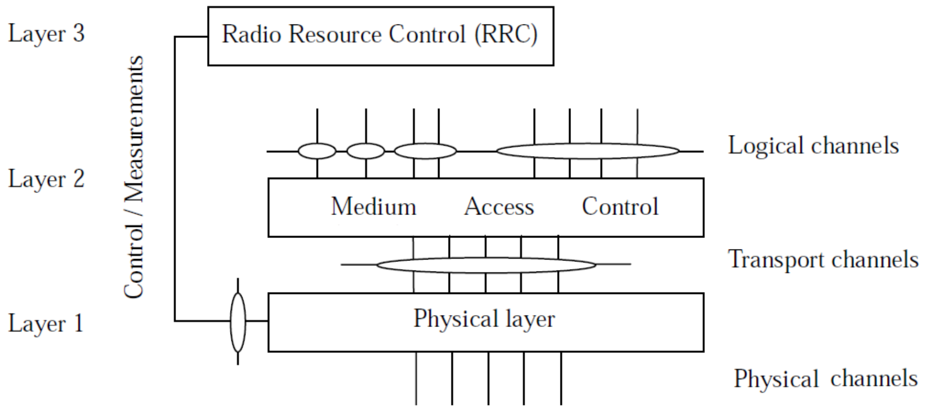


Figure 2.8: Radio interface protocol architecture around the physical layer (adapted from [3GP11a]).

associated with a Service Access Point (SAP) between different layers (the circles in Figure 2.8). Logical channels provide services at the SAP between the MAC and RLC layers and are characterized by the type of information they carry. Transport channels provide services at the SAP between the MAC and PHY layers and are characterized by how the information is transferred over the radio interface. Physical channels are the actual implementation of transport channels over the radio interface, carrying control messages and user data.

2.3.1 Interface Protocols

Each interface in the network uses standard Internet Engineering Task Force (IETF) transport protocols, which are shown in Figure 2.9. Unlike the air interface, these interfaces use protocols from Layers 1 to 4 of the usual Open Systems Interconnection (OSI) model.

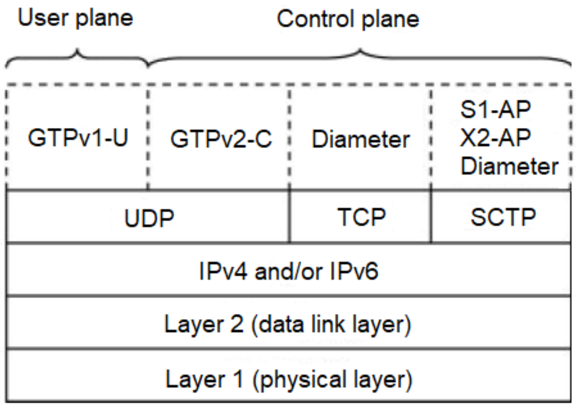


Figure 2.9: Transport protocols used by the network (adapted from [Chr12]).

At the bottom of the protocol stack, the transport network can use any suitable protocols for L1 and L2, such as Ethernet. Every network element is then associated with an IP address and the network uses the IP to route information from one element to another across the underlying transport network.

Above IP, three transport layer protocols are used. User Datagram Protocol (UDP) transmits data packets from one network element to another, being the only transport protocol used in the user plane. Transmission Control Protocol (TCP) re-transmits packets if they arrive incorrectly. Stream Control Transmission Protocol (SCTP) is similar to TCP, but includes additional features that make it more suitable for the delivery of signalling messages. The control plane chooses its transport protocol depending on the overlying signalling protocol.

The LTE user plane contains mechanisms to forward data correctly between the UE and the P-GW, as well as to quickly respond to changes in the UE's location. These mechanisms are implemented by a 3GPP protocol known as the GPRS Tunelling Protocol-User plane (GTPv1-U) and are used over the S1-U, X2 and S5/S8 interfaces (shown in Figure 2.4). GTPv1-U forwards packets from one network element to another using a technique known as tunnelling.

LTE uses several signalling protocols. On the air interface, the eNB controls an UE's radio communications by using signalling messages written using the RRC protocol, over the Uu interface. In the RAN, the serving MME controls the eNBs within its pool area using the S1 Application Protocol (S1AP) over the S1-MME interface, while the eNBs communicate with each other using the X2 Application Protocol (X2AP) X2 interface. If the access network is composed of a single eNB, no X2 interface is required.

Inside the EPC, the HSS and MME communicate over the S6a interface using the Diameter protocol, a standard protocol for authentication, authorization and accounting.

The interfaces S5/S8 and S11 utilize a 3GPP protocol known as the GPRS Tunelling Protocol-Control plane (GTPv2-C). This protocol allows for peer-to-peer communications between the different elements of the EPC and is responsible for managing the GTPv1-U tunnels that were previously mentioned in this section [Chr12].

2.4 Physical Layer

In this section, LTE's physical layer is described. The multiple access technologies used on the downlink and uplink are discussed, followed by the radio frame structure and resource grid, ending with the physical channels and signals.

2.4.1 Multiple Access Technologies

Downlink and uplink transmissions in LTE are based on the use of multiple access technologies: specifically, Orthogonal Frequency Division Multiple Access (OFDMA) for the downlink and Single-Carrier Frequency Division Multiple Access (SC-FDMA) for the uplink. Both will be described in the following sections.

2.4.1.1 Downlink: Orthogonal Frequency Division Multiple Access (OFDMA)

Orthogonal Frequency Division Multiple Access (OFDMA) is a variant of Orthogonal Frequency Division Multiplexing (OFDM), a digital multi-carrier modulation scheme that is widely used in wireless systems and more recently in cellular. Instead of transmitting an high-rate data stream using a single carrier, OFDM makes use of a large number of orthogonal subcarriers that are transmitted in parallel. These subcarriers are closely spaced but do not interfere with one another in the frequency domain. Each subcarrier is modulated with a conventional modulation scheme at a low symbol rate, such as Quadrature Phase Shift Keying (QPSK), 16 or 64-Quadrature Amplitude Modulation (QAM), described later in section 2.5. The combination of a great number of subcarriers using lower bandwidths each enables similar data rates to those obtained when using conventional wideband single-carriers with higher bandwidths.

The diagram in Figure 2.10 illustrates the key features of an OFDM signal in frequency and time. In the frequency domain, multiple subcarriers are each independently modulated with data. In the time domain, the long symbols used for OFDM are separated by a guard interval known as Cyclic Prefix (CP), to improve the resilience of the system. The CP is a copy of the ending part of a symbol that is inserted at its beginning. The receiver uses this guard interval to avoid the Inter-Symbol Interference (ISI) between adjacent symbols, caused by multipath reflection delay spread, by sampling the received waveform at the optimum time.

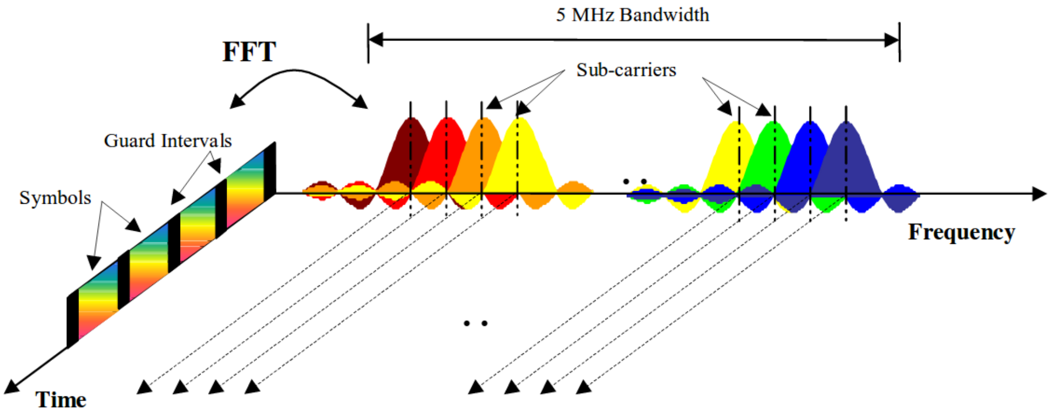


Figure 2.10: Frequency-Time Representation of an OFDM signal (taken from [3GP04]).

Nevertheless, OFDM has some disadvantages. Because the subcarriers are tightly spaced, OFDM is more easily affected by frequency errors and phase noise, causing the subcarriers to start losing their orthogonality. OFDM also creates high Peak-to-Average Power Ratio (PAPR) signals which can be problematic to amplifiers, increasing the power consumption.

All subcarriers in OFDM are attributed to a single user at the same time, making only one user able to transmit at a time. If more than one user is trying to transmit using OFDM, they have to take turns. With OFDMA, however, the subcarriers are directly assigned in frequency to different users. That is why, for the downlink,3GPP chose OFDMA.

The result is a more robust system with increased capacity. OFDMA can adjust the modulation and coding for each subcarrier, it has better spectral efficiency and low-complexity modulation [Yan10]. As the users are scheduled by frequency, frequency-selective fading is less prone to happen [Agi09]

A simplified block diagram for the signal generation and reception of OFDMA is illustrated in Figure 2.11. It begins with the mapping of M data bits to their respective modulations, and then the modulated data symbols into the available subcarriers. An Inverse Fast Fourier Transform (IFFT) is performed to convert the data symbols to the time domain, where a CP is inserted for robustness of the system, preventing multipath fading. After going through the air interface, the symbols enter the receiver and go through a reverse process, which ends in their de-mapping back to M data symbols. Time and frequency synchronization must be accurate, especially for the CP removal step, otherwise a wrong part of the symbol will be dropped.

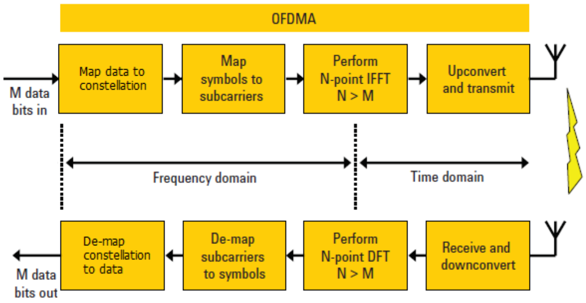

OFDMA
M data bits in
Map data to constellation
Map symbols to subcarriers
Perform N-point IFFT N > M
Upconvert and transmit
Frequency domain
Time domain
M data bits out
De-map constellation to data
De-map subcarriers to symbols
Perform N-point DFT N > M
Receive and downconvert

Figure 2.11: Simplified model of OFDMA signal generation and reception (adapted from [Agi09]).

2.4.1.2 Uplink: Single-Carrier Frequency Division Multiple Access (SC-FDMA)

The high PAPR associated with OFDM led 3GPP to look for a different transmission scheme for the LTE uplink. SC-FDMA was chosen because it combines the low PAPR techniques of single-carrier transmission systems, such as GSM, with the multipath fading resistance and flexible frequency allocation of users of OFDMA.

The block diagram for the signal generation and reception of SC-FDMA is illustrated in Figure 2.12. SC-FDMA signal generation begins with a special precoding process but then continues in a identical manner to OFDMA. SC-FDMA data symbols in the time domain are converted to the frequency domain using a Discrete Fourier Transform (DFT). Then, in the frequency domain, they are mapped to the desired location in the channel bandwidth before being converted back to the time domain, using an IFFT. Finally, the CP is inserted, to provide robustness to the system against multipath.

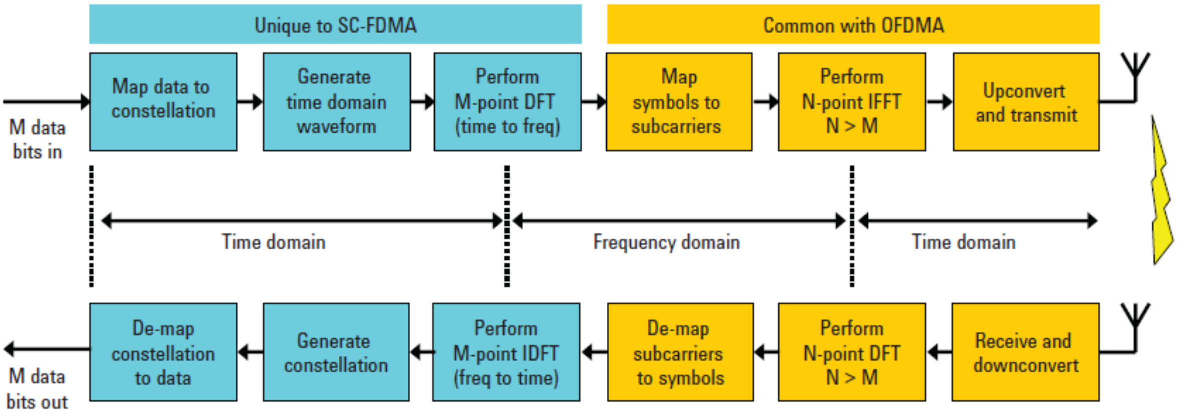
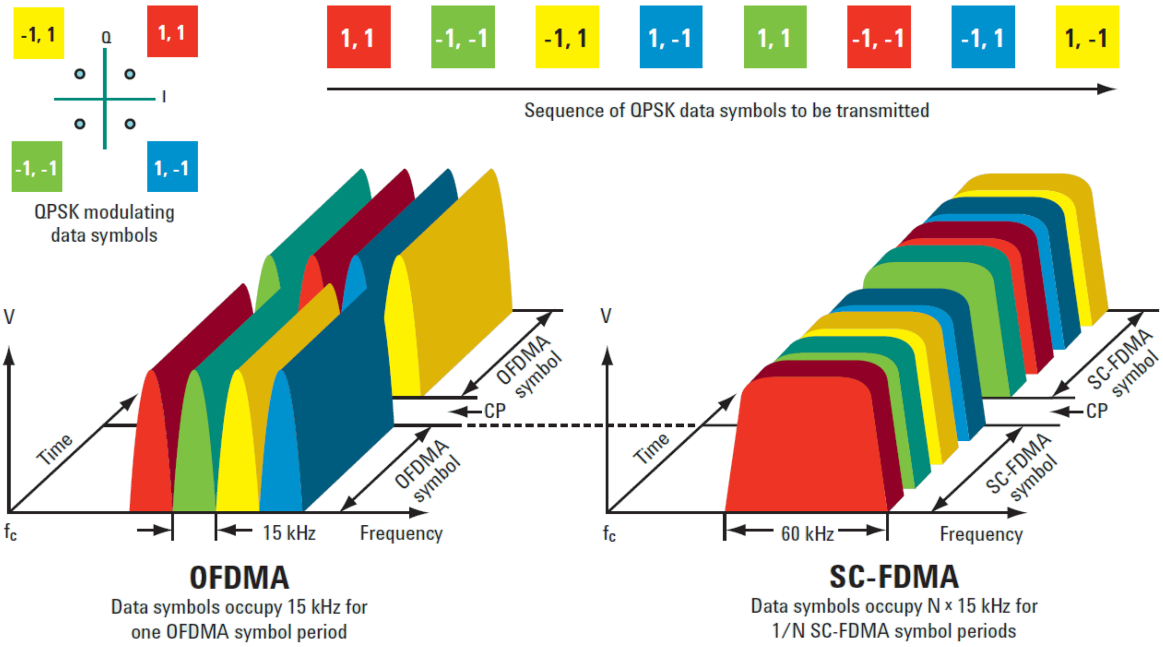

Unique to SC-FDMA
Common with OFDMA
M data bits in
Map data to constellation
Generate time domain waveform
Perform M-point DFT (time to freq)
Map symbols to subcarriers
Perform N-point IFFT N > M
Upconvert and transmit
Time domain
Frequency domain
Time domain
M data bits out
De-map constellation to data
Generate constellation
Perform M-point IDFT (freq to time)
De-map subcarriers to symbols
Perform N-point DFT N > M
Receive and downconvert

Figure 2.12: Simplified model of SC-FDMA signal generation and reception (taken from [Agi09]).

For the reception of the SC-FDMA signal, the process follows the same steps as for OFDMA, with the addition of performing an Inverse Discrete Fourier Transform (IDFT) that converts the frequency-shifted signal to the time domain, along with the rest of the decoding process related to SC-FDMA.

A graphical comparison of OFDMA and SC-FDMA is shown in Figure 2.13. The CP is shown as a gap; however, it is actually filled with a copy of the end of the next symbol.



-1, 1
Q
1, 1
I
-1, -1
1, -1
QPSK modulating data symbols
1, 1
-1, -1
-1, 1
1, -1
1, 1
-1, -1
-1, 1
1, -1
Sequence of QPSK data symbols to be transmitted
V
Time
fc
15 kHz
Frequency
OFDMA symbol
CP
OFDMA symbol
OFDMA
Data symbols occupy 15 kHz for one OFDMA symbol period
V
Time
fc
60 kHz
Frequency
SC-FDMA symbol
CP
SC-FDMA symbol
SC-FDMA
Data symbols occupy N x 15 kHz for 1/N SC-FDMA symbol periods

Figure 2.13: Comparison of OFDMA and SC-FDMA transmitting a series of QPSK data symbols (taken from [Mor08]).

As illustrated, OFDMA transmits the four QPSK data symbols in parallel, one per subcarrier, while SC-FDMA transmits them in series at four times the rate, meaning each data symbol occupies four times the OFDMA's symbols bandwidth.

It is the parallel transmission of multiple symbols that creates the undesirably high PAPR of OFDMA. By transmitting N data symbols in series at N times the rate, the SC-FDMA occupied bandwidth is the same as multi-carrier OFDMA but, most importantly, the PAPR is the same as that used for the original data symbols. As the number of subcarriers increases, the PAPR of OFDMA with random modulating data approaches Gaussian noise statistics, but the SC-FDMA PAPR remains the same as that used for the original data symbols [Agi09].

In each transmission, a scheduling decision is made where each scheduled UE is assigned a certain amount of radio resources in the time and frequency domain [Aru11]. In the next section, the frame structure and radio resource allocation is described.

2.4.2 Frame Structure

The frame structure in the time domain is a common element shared by both downlink and uplink. In LTE specifications, the size of various fields in the time domain is expressed as a number of time units $T_s = 1/(15000 \times 2048)$ seconds. As the subcarrier spacing is defined to be $\Delta_f = 15$ kHz, T_s can be regarded as the sampling time of a Fast Fourier Transform (FFT)-based OFDM/SC-FDMA transmitter/receiver implementation with a maximum FFT size of 2048, being equivalent to a sampling frequency of 30.72 MHz. This sampling frequency corresponds to a channel bandwidth of 20 MHz. Other FFT sizes are supported and their relation with the channel bandwidth and sampling frequency is specified in Table 2.1 [Aru11].

Channel bandwidth [MHz]	1.4	3	5	10	15	20
Sampling frequency [MHz]	1.92	3.84	7.68	15.36	23.04	30.72
Number of RBs	6	15	25	50	75	100
FFT size	128	256	512	1024	1536	2048
Subcarrier Spacing	15 kHz					
RB Bandwidth	180 kHz					

Table 2.1: Transmission parameters for LTE (adapted from [Aru11]).

Downlink and uplink transmissions are organized into radio frames with duration $T_f = 307200 \times T_s = 10$ ms. For flexibility, LTE supports both FDD and TDD modes. Accordingly, two kinds of radio frame structures are supported: type 1, applicable to FDD, and type 2, applicable to TDD [3GP13], described in the following sections.

2.4.2.1 Frame structure type 1 - Frequency Division Duplex (FDD)

Frame structure type 1 is applicable to both full-duplex and half-duplex Frequency Division Duplex (FDD). There are three different kinds of units specified for this frame structure, pictured in Figure 2.14. The smallest one is called a slot, which is of length $T_{slot} = 15360 \times T_s = 0.5$ms. Two consecutive slots are defined as a subframe of length 1 ms and 20 slots constitute a radio frame of 10 ms.

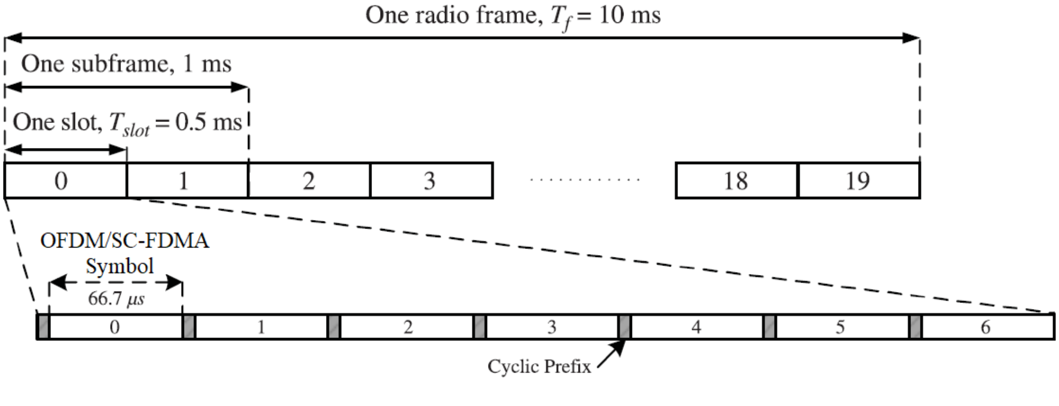


Figure 2.14: Frame structure type 1 (adapted from [Aru11]).

Each slot consists of a number of OFDM/SC-FDMA symbols - for the downlink and uplink, respectively - including CPs. As discussed previously, the CP is a guard interval used to eliminate the ISI and should be larger than the channel delay spread. Therefore, its length depends on the environment the network operates in. As can be seen in Figure 2.14, the symbol time is $1/\Delta_f \approx 66.7\mu s$, considering a subcarrier spacing of 15 kHz. The normal CP length corresponds to seven symbols per slot. It is suitable for urban environments and high data rate applications. There is also an extended CP, which only allows six symbols per slot, but it is not relevant for this dissertation work.

For FDD, uplink and downlink transmissions are separated in the frequency domain, each with 10 subframes. This grants the use of different ranges of frequencies for the UL and DL when using FDD LTE frequency bands, explained in section 2.6. In half-duplex FDD operation, the UE cannot transmit and receive at the same time, while there are no such restrictions in full-duplex FDD [3GP13]. However, full-duplex FDD terminals need high quality and expensive RF duplex-filters to separate uplink and downlink channels, while half-duplex FDD allows for the hardware to be shared between the uplink and downlink, reducing the data rates by half but costing less [Aru11].

2.4.2.2 Frame structure type 2 - Time Division Duplex (TDD)

Frame structure type 2 is applicable to Time Division Duplex (TDD). As shown in Figure 2.15, each radio frame of length $T_f = 307200 \times T_s = 10$ms consists of two half-frames of length $153600 \times T_s = 5$ms each. Each half-frame consists of five subframes of length $30720 \times T_s = 1$ms.

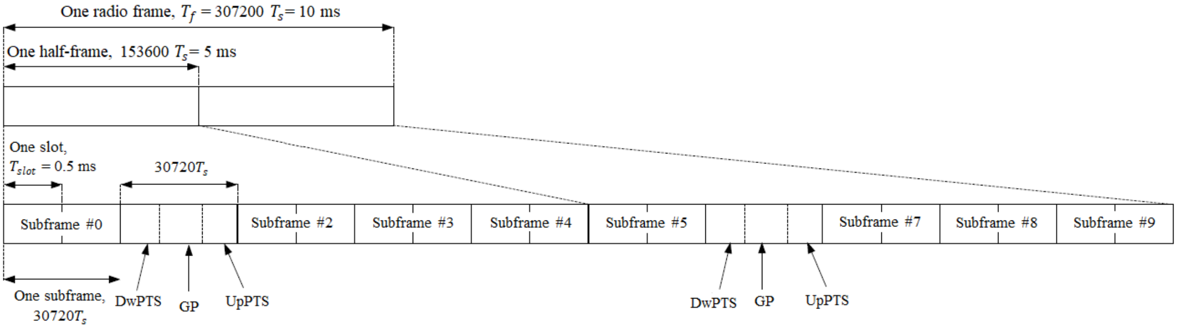


Figure 2.15: Frame structure type 2 (adapted from [3GP13]).

With TDD, the LTE frequency bands (described in section 2.6) operate in the same range of frequencies for both uplink and downlink, as the 20 subframes are transmitted altogether. Only a subset of the subframes are available for downlink transmission ('D'). The remaining subframes are reserved for uplink transmission ('U') and for special subframes ('S'), with the latter being divided in three fields: Downlink Pilot Time Slot (DwPTS) and Uplink Pilot Time Slot (UpPTS), separated by a Guard Period (GP), to allow for switching between downlink and uplink transmission [Ste11]. The allocation of these subframes can be one of the seven different defined uplink/downlink configurations, in Table 2.2.

The amount of special subframes depends on the downlink-to-uplink switch-point periodicity: in case of 5ms, the special subframe exists in both half-frames; in case of 10 ms, it exists in the first half-frame only. Subframes 0 and 5 and DwPTS are always reserved for downlink transmission. UpPTS and the subframe immediately following the special subframe are always reserved for uplink transmission [3GP13].

Table 2.2: Uplink/Downlink configurations for TDD: 'D' for the downlink subframes, 'U' for the uplink subframes and 'S' for the special subframes (taken from [3GP13]).

Uplink-Downlink configuration	**Downlink-to-Uplink Switch-point periodicity**	**Subframe number**									
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	5 ms	D	S	U	U	U	D	S	U	U	U
1	5 ms	D	S	U	U	D	D	S	U	U	D
2	5 ms	D	S	U	D	D	D	S	U	D	D
3	10 ms	D	S	U	U	U	D	D	D	D	D
4	10 ms	D	S	U	U	D	D	D	D	D	D
5	10 ms	D	S	U	D	D	D	D	D	D	D
6	5 ms	D	S	U	U	U	D	S	U	U	D

2.4.2.3 Resource Grid

The physical resource in the downlink/uplink for each slot is described by a time-frequency grid, called a resource grid, as illustrated in Figure 2.16. Each column and each row of the resource grid correspond to one OFDM/SC-FDMA symbol and one OFDM/SC-FDMA subcarrier, respectively. The duration of the resource grid in the time domain corresponds to one slot in a radio frame. The smallest time-frequency unit in a resource grid is denoted as a resource element.

Each resource grid consists of a number of RBs (N_{RB}), which are the basic elements for radio resource allocation [Aru11]. As specified in Table 2.3, a RB is composed by 6 or 7 symbols (N_{symb}) depending on the CP configuration and 12 consecutive subcarriers (N_{SC}). Each RB is 180 kHz wide in the frequency domain ($N_{SC} \times \Delta_f = 180$ kHz) and is of equal size as one slot in the time domain, 0.5 ms.

There are $N_{symb} \times N_{SC}$ resource elements inside a RB. Therefore, there are $N_{RB} \times N_{symb} \times N_{SC}$ resource elements inside a resource grid. Table 2.1 lists the channel bandwidths and their relation with the number of RBs.

Configuration	N_{SC}	N_{symb}
Normal CP (Δ_f = 15kHz)	12	7
Extended CP (Δ_f = 15kHz)	12	6

Table 2.3: RB parameters

2.4.3 Physical Channels and Signals

Each physical channel corresponds to a set of resource elements in the time-frequency grid that carry information from higher layers. The basic entities that make a physical channel are resource elements and RBs, described in the previous section. Besides physical channels, there are physical signals embedded in the downlink and uplink physical layer, which do not carry information from higher layers. The physical signals defined in the LTE specifications support the lowest-level operation of the PHY layer and are separated in reference signals and synchronization signals.

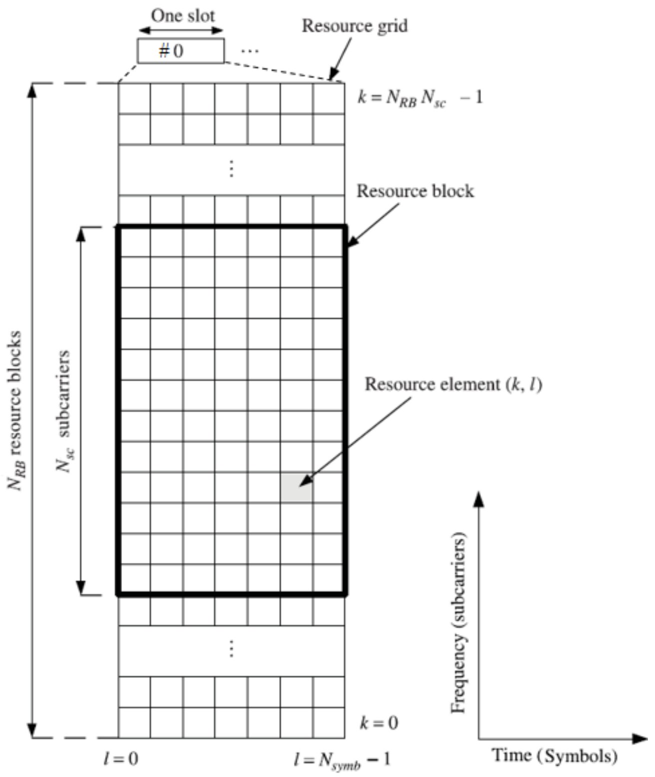


Figure 2.16: Structure of the resource grid (adapted from [Aru11]).

2.4.3.1 Downlink Physical Channels

The main LTE downlink physical channels are the data channel Physical Downlink Shared Channel (PDSCH), the control channels Physical Downlink Control Channel (PDCCH), Physical Control Format Indicator Channel (PCFICH) and, finally, Physical Hybrid-ARQ Indicator Channel (PHICH). There is also a broadcast channel, the Physical Broadcast Channel (PBCH). These are all described below [Eri16, Aru11]:

- Physical Downlink Shared Channel (PDSCH): carries user data and paging information.
- Physical Downlink Control Channel (PDCCH): carries the Downlink Control Information (DCI) and is used for reception of PDSCH, as well as for scheduling grants enabling transmission on the Physical Uplink Shared Channel (PUSCH). The DCI contains information regarding the transport format, resource allocation and HARQ.
- Physical Control Format Indicator Channel (PCFICH): provides the UEs with information necessary to decode the set of PDCCHs, such as the number of OFDM symbols used. This channel carries the Control Format Indicators (CFIs), which tells the devices about the organization of data and control information on the downlink.
- Physical Hybrid-ARQ Indicator Channel (PHICH): carries the Hybrid-ARQ Indicators (HIs) associated with uplink data transmissions, to indicate to the device whether a transport block should be retransmitted or not.

- Physical Broadcast Channel (PBCH): carries the Master Information Block (MIB), required by the UE in order to access the network. The MIB contains information about the system's bandwidth (see table 2.1) and the PHICH configuration.

2.4.3.2 Uplink Physical Channels

The three main uplink physical channels are the PUSCH, Physical Uplink Control Channel (PUCCH) and Physical Random Access Channel (PRACH), detailed below [Eri16, Aru11]:

- Physical Uplink Shared Channel (PUSCH): it is the uplink counterpart of the PDSCH. There is, at most, one PUSCH per uplink component carrier per device [Chr12].

- Physical Uplink Control Channel (PUCCH): carries Uplink Control Information (UCI) including Channel State Information (CSI), aiding downlink channel-dependent scheduling, HARQ acknowledgements in response to downlink transmissions on the Downlink Shared Channel (DL-SCH) and uplink scheduling requests, if the UE wishes to transmit uplink data on the PUSCH but does not have resources to do so. There is only one PUCCH per device.

- Physical Random Access Channel (PRACH): carries the random access preamble sent by the UEs.

2.4.3.3 Reference Signals

Reference Signals (RSs) are predefined signals occupying specific resource elements within the time-frequency grid, whose job is to support accurate channel estimation for the demodulation of physical channels. Each RS pattern is transmitted from an antenna port at the eNB [Ste11]. The main LTE downlink reference signals are Cell-Specific Reference Signal (CRS), DeModulation Reference Signal (DMRS), Channel State Information Reference Signal (CSI-RS) and Positioning Reference Signal (PRS), while the two uplink reference signals are the Uplink DMRS and Sounding Reference Signal (SRS). All of these reference signals are described below [Eri16, Aru11]:

- Cell-Specific Reference Signal (CRS): used for cell search, downlink channel quality measurements and estimation.

- DeModulation Reference Signal (DMRS): also known as *UE-specific reference signal*, is used for channel estimation by a specific UE.

- Channel State Information Reference Signal (CSI-RS): used by devices to acquire CSI.

- Positioning Reference Signal (PRS): estimates the geographical position of the UE.

- Uplink DMRS: used for data reception, associated with transmissions of uplink data on the PUSCH and/or control signalling on the PUCCH.

- Sounding Reference Signal (SRS): used for channel quality determination to enable frequency-selective scheduling on the uplink.

2.4.3.4 Synchronization Signals

The cell search procedure in LTE begins with a synchronization procedure which makes use of two specially designed physical signals that are broadcast in each cell: the Primary Synchronization Signal (PSS) and the Secondary Synchronization Signal (SSS). The detection of these two signals not only enables time and frequency synchronization, but also provides the UE with the physical layer identity of the cell and the Cyclic Prefix (CP) length, and informs the UE whether the cell uses FDD or TDD [Ste11].

2.5 Modulation Schemes

The modulation schemes supported by LTE are Binary Phase Shift Keying (BPSK), QPSK, 16-QAM and 64-QAM, being that the first can't be used for downlink and uplink data transmissions, but rather for a limited number of control streams (e.g. PHICH). On the other hand, PUSCH, for example, uses the last three modulation schemes. Their constellation diagrams are illustrated in Figure 2.17.

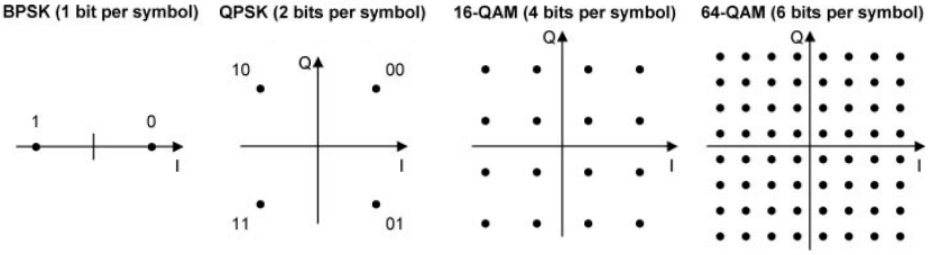


Figure 2.17: Constellations of the modulation schemes support by LTE (taken from [Ste11]).

BPSK sends 1 bit per modulated symbol at a time, using two states located at of 0°and 180°. QPSK takes 2 bits per symbol at a time and transmits them using a radio wave that can have four different states. These correspond to bit combinations of 00, 10, 11 and 01. 16-QAM sends 4 bits per symbol at time, using 16 states that have different amplitudes and phases. Similarly, 64-QAM sends 6 bits per symbol at a time using 64 different states, so it has a data rate six times greater than that of BPSK [Chr12].

The modulation schemes used on a certain channel are chosen by taking into account some basic criteria: Bit Error Rate (BER), Signal-to-Noise Ratio (SNR), available bandwidth, power efficiency, better QoS and cost effectiveness. Modulation methods which are capable of transmitting more bits per symbol are also more vulnerable to errors caused by noise [Cha15].

2.6 Spectrum and Frequency Bands

3GPP specifications allow for the deployment of LTE in a wide variety of spectrum bands, globally. It is deployable in any of the existing 2G and 3G spectrum bands, as well as several new frequency bands that may be identified [Aru11], in both licensed and unlicensed spectrum. Operating in different frequency bands has limited or no impact on a radio access technology perspective. What may differ between them are mainly the more specific RF requirements, such as the allowed maximum transmit power, noise on the air interface, requirements/limits on Out-of-Band (OOB) emission and so on.

The frequency bands are divided into paired bands (FDD), where separated frequency ranges are assigned for uplink and downlink, and unpaired bands (TDD), with a single shared frequency range, as was previously explained. Release 14 includes a total of 37 frequency bands for FDD and 16 for TDD. Table 2.4 specifies some of the frequency bands used in LTE.

E-UTRA Operating Band	**UL Range [MHz]**	**DL Range [MHz]**	**Duplex Mode**
1	1920 - 1980	2110 - 2170	FDD
3	1710 - 1785	1805 - 1880	FDD
5	824 - 849	869 - 894	FDD
7	2500 - 2570	2620 - 2690	FDD
8	880 - 915	925 - 960	FDD
20	832 - 862	791 - 821	FDD
22	3410 - 3490	3510 - 3590	FDD
31	452.5 - 457.5	462.5 - 467.5	FDD
38	2570 - 2620	2570 - 2620	TDD
40	2300 - 2400	2300 - 2400	TDD

Table 2.4: Some of the frequency bands used in LTE (adapted from [3GP17a]).

Due to the propagation properties, lower-frequency bands (e.g. Band 31) are good for wide-area coverage deployments, both in urban, suburban and rural environments. Higher-frequency bands, on the contrary, are more difficult to use for wide-area coverage, having therefore been used for boosting capacity in dense deployments. With new services requiring higher data rates and higher capacity in dense deployments, frequency bands above 6 GHz are being looked at as a complement to the frequency bands below 6 GHz. With the upcoming fifth generation of telecommunications requirements for extreme data rates and localized areas with very high area traffic capacity demands, deployments using much higher frequencies (even above 60 GHz) is considered [Eri16].

This chapter contains introductory concepts regarding the LTE standard, providing a basis for the dissertation work. First, the network architecture was described, specifying the functionalities of the UE, E-UTRAN and EPC, along with their respective nodes and identifiers. Then, the LTE protocol stack was presented, followed by a deeper description of the physical layer, where aspects such as the frame structures used for the TDD and FDD duplex modes, LTE physical channels, multiple access and modulation schemes were approached. To finalize, a selection of LTE frequency bands available for transmission were presented.

The next chapter will provide an overview of several experimentation frameworks that implement the LTE standard by means of software, instead of the traditional hardware implementations. The main objective is to pick the most adequate platform to be used for experimentation on this dissertation. To be run along with the chosen software framework, several software radio hardware platforms will be compared.

Chapter 3

LTE Experimentation Frameworks

This chapter begins with an introduction, in section 3.1, followed by an overview and comparison of several experimentation frameworks that implement the LTE standard, in section 3.2. One of the analysed software platforms is selected as a point of departure for the implementation of the proposed dissertation work. That choice is detailed in section 3.3.

3.1 Introduction

Testbeds are essential to experimentally evaluate new technologies. Their architecture should be flexible, so it is possible to easily update the system for certain configurations. Scalability is an important factor, meaning that various configurations with different hardware capability requirements are supported. Furthermore, the testbed should support logging, so the user can analyse the system's output and performance.

An LTE-specific testbed is usually composed by one or more UEs, an eNB and an EPC. A wide variety of COTS hardware products available nowadays already provide a solution for each of these components. Albeit providing an implementation with very few to no errors, these commercial products are too complex (or sometimes impossible) to modify and to add new features. As doing so could prove costly, open-source LTE SDR software is preferred for research and prototype developing purposes. Before going into more details, the concept of software-defined radio will be explained.

Software-Defined Radio (SDR) is a concept in which radio communication systems' components are implemented by means of software, instead of the traditional hardware implementation. As most communication modules are implemented by software (e.g. frequency band selection, modulation/demodulation, filtering, encoding/decoding, signal enhancement), the technology upgrading process is more flexible and less expensive, because the user only has to upgrade the software and can keep the hardware as is. This kind of flexibility allows for operation on multi-band and multi-modulation [Tor10]. SDR systems can be implemented on various reconfigurable hardware platforms, such as Field-Programmable Gate Arrays (FPGAs), Application-Specific Integrated Circuits (ASICs), Digital Signal Processors (DSPs) or GPPs. When compared to the others, GPP-based SDR systems have advantages in reconfigurability and flexibility because high-level programming languages are used, which are cross-platform, instead of the hardware description languages used on the former - e.g. Very High Speed Integrated Circuit Hardware Description Language (VHDL).

An usual GPP-based SDR system consists of a peripheral equipment connecting to the

GPP; the first being responsible for the frequency conversion and digitization and the latter for baseband signal processing [Hen16]. Several of these peripheral equipments will be discussed through the course of this dissertation's work.

Several experimentation frameworks that implement the LTE elements - UE, eNB and EPC - are evaluated in the next section.

3.2 Evaluation of LTE implementation platforms

This dissertation's goal is to build a fully LTE standard-compliant real-time platform testbed, with special attention to the UE. There are several open-source software platforms available, each with different characteristics and implementations in either simulation or real-time mode. For the purpose of this dissertation's work, simulation is not relevant, so, to fulfil its goal, this platform must be able to run with the aid of commercial SDR hardware like the previously mentioned USRP. The framework should offer realism regarding the internal structure of the EPS, including the authentication and key agreement and an UE running in SDR. Logging is required as well, to allow for easier debugging and understanding of the system. The following sections are dedicated to each analysed platform.

3.2.1 gr-LTE

gr-LTE [gr-13] is an open-source software package which aims to provide a GNU Radio LTE Receiver to receive, synchronize and decode LTE signals. It can basically be considered as an UE. While gr-LTE is nice for familiarizing with the LTE protocol, it isn't fully developed and doesn't provide an eNB or EPC implementation. Thus, it cannot be used for the purpose of this dissertation.

3.2.2 OpenEPC

OpenEPC [Ope17b] provides a full implementation of the LTE Release 12 core network and includes an emulation of the eNB and UE. The name is deceiving though, as the source code is only available under paid licensing models. Therefore, this platform was not chosen.

3.2.3 Open-Source LTE Deployment (OSLD)

The mission of the Open-Source LTE Deployment (OSLD) project [Ope13] is to promote open-source SDRs and shared development of software for wireless communications systems. It provides a basic library of LTE PHY layer DSP blocks, implemented in a general-purpose way for multi-platform use. It relies on Abstraction Layer and Open Operating Environment++ (ALOE++) [ALO12], an open-source middleware and execution environment for distributed signal processing, and provides two main waveform description files, one for the LTE DL and one for the UL.

It can be ran in an USRP although a Real-Time Operating System (RTOS) is necessary for latency reduction, otherwise it won't work. As no EPC implementation is provided and there isn't an active community anymore, this platform was not chosen.

3.2.4 ns-3 with the LTE Module

ns-3 [ns-17] is an open-source discrete-event network simulator targeted primarily for research and educational use, written in C++. Its LTE Module contains software libraries that allow the simulation of LTE networks and the EPC. It implements the entire LTE radio protocol stack, whose entities reside within the eNB and UE nodes, and a partial core network with S-GW, P-GW and MME nodes (missing several HSS functionalities).

While the code is extensively documented and the community is active, it only simulates the radio communication. As using physical radio hardware is required in this dissertation, this platform was not chosen.

3.2.5 OpenLTE

OpenLTE [ope17c] is an open-source implementation of the 3GPP LTE specifications, written in Octave, Python and C++. The project's current focus is on adding capabilities to its simple eNodeB application and extending the capabilities of the GNU Radio applications. These include LTE In-phase/Quadrature (I/Q) file recording or transmission and reception of downlink communications, using either of the following peripheral SDRs: rtl-sdr [RTL17], HackRF One [Hac17] or USRP B2X0 [Ett17].

OpenLTE partly implements the MME and HSS, including part of the authentication procedure. There's no UE implementation, however, and many other features are still unstable or under development. It also requires a great amount of processing power, as well as very low latency. If there is any delay in the processing, the system won't be able to respond in time and will lose samples.

The project's community isn't very active nowadays. Despite of not providing an UE implementation, OpenLTE won't be discarded as an option, for now.

3.2.6 srsLTE, srsUE and srsENB

srsLTE is an open-source LTE library for the PHY layer, written in C, which includes srsUE and srsENB [SRS17], a complete software radio LTE UE and eNB, respectively, written in C++.

srsLTE, developed by Software Radio Systems, provides a LTE Release 8 compliant implementation, for FDD, including all DL and UL channels/signals. The code is well-structured and divided in modules, allowing for easier customization or replacement of components without affecting the existing parts.

srsUE covers all UE layers from PHY to IP and builds upon the srsLTE library which provides the PHY layer processing [Ism16]. srsENB also builds upon the srsLTE library but requires a commercial license, so it won't be used. The full EPS protocol stack is supported and some functions are imported from the OpenLTE project.

Regarding hardware, srsLTE is able to work with any RF front-end and currently gives support to the Ettus USRP B210/X300 and Nuand's bladeRF [Nua17]. srsUE has been tested and validated with several commercial eNBs and also works with other open-source platforms, so this option won't be discarded.

3.2.7 AMARI LTE 100

Amarisoft's AMARI LTE 100 [Ama17] requires a paid license to use, but it is arguably the best and most complete LTE SDR implementation available. It's a software suite which provides an LTE Release 13 compliant eNB, EPC, Evolved Multimedia Broadcast Multicast Services (eMBMS) gateway and IMS server, allowing end-to-end communication with up to 1000 commercial UEs. There's also AMARI UE 100, a UE simulator software that simulates up to 1000 UEs.

It can be built in a SDR PCIe board [Ama17], USRP N-series or LimeSDR [Lim17]. As it is not an open-source platform, it was not picked.

3.2.8 OpenAirInterface

OpenAirInterface (OAI) wireless technology platform is a flexible platform towards an open LTE ecosystem, written in C. It offers an open-source software-based implementation of the LTE system, spanning the full protocol stack of the 3GPP standard both in E-UTRAN and EPC. It can be used to build and customize a LTE base station, a user equipment and a core network on a Linux-based computer (Intel x86 processors).

The eNB can be connected either to commercial UEs or OAI UEs to test and monitor different configurations and network setups in real-time [Ope17a]. It works with several hardware RF front-ends, such as the USRP B210, as well as in simulation mode, where the radio interface is simulated by Ethernet. Several built-in tools are included, such as a protocol analyzer, a configurable logging system for all layers and channels plus debugging and soft monitoring tools.

While OAI is a very complete implementation, it is also very complex. The code is at times confusing and not well organized and documented, making it hard to understand or customize for a new user. It is in constant development, so it might also prove to be unstable. The community is very active, which can be helpful for problem solving. This was the last platform to be analysed and it will also be considered, leaving us with three open-source platforms to choose from.

3.2.9 Final decision

Eight software radio platforms were analysed, in which three stood out: OpenLTE, srsLTE and OAI. As srsLTE can only be used to implement a user equipment (srsUE), the first decision to be made is between the OpenLTE and OAI platforms. OpenLTE's code is well structured and documented, however it is missing many features. OAI, on the other hand, is an almost complete implementation and offers great performance when well configured. Therefore, OpenLTE is discarded and OAI is picked as the main platform.

As srsUE is still in play, the second decision relies on choosing which SDR UE is going to be implemented, srsUE or OAI UE. Over the past year OAI UE had serious improvements and its characteristics are identical to the srsUE's, with both achieving the same Internet speeds in similar circumstances. Therefore, in other to avoid compatibility issues and deep multi-platform knowledge, srsLTE is discarded.

Regarding the other platforms, either they weren't open-source or didn't have enough capabilities for the purpose of this dissertation's work, so they weren't considered. As the selected platform, the following section will be entirely dedicated to OAI.

3.3 OpenAirInterface (OAI)

OpenAirInterface (OAI) is part of an alliance established in 2014, the OpenAirInterface Software Alliance (OSA), whose purpose is to provide software and tools for 5G wireless research and product development. OSA is a French non-profit organization, funded by corporate sponsors and founded by EURECOM, a French graduate school and research center in communication systems. The alliance promotes a meritocratic process in which individual members can contribute to the OSA software development of both the core software or the projects run by various member organizations.

OSA currently provides a 3GPP standard-compliant software implementation of a subset of LTE's Release 10 for the UE, eNB, MME, HSS, S-GW and P-GW on standard Linux-based computing equipment (Intel x86 PC/ARM architectures). Figure 3.1 represents the network entities that are currently implemented by OAI. The Policy Control and Charging Rules Function (PCRF) node is missing, even though its functionalities are coded in other network entities. Interoperability with the previous network generations is currently not possible, as the nodes Radio Network Controller (RNC), Serving GPRS Support Node (SGSN) and Equipment Identity Register (EIR) are also missing. The S-GW and P-GW are actually combined in a single node, thus eliminating the S5/S8 interfaces.

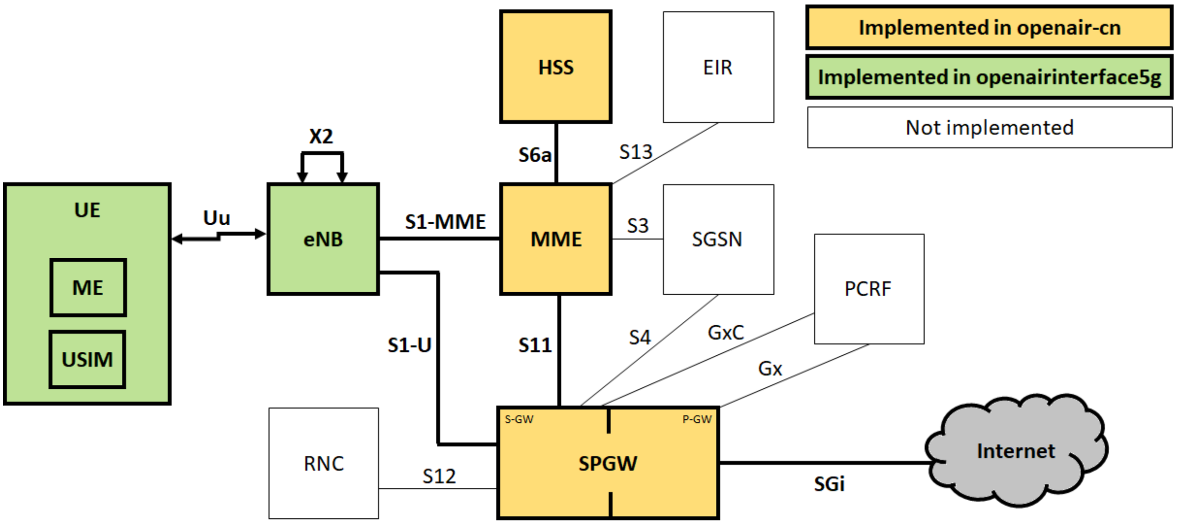


Figure 3.1: LTE network entities implemented by OpenAirInterface (adapted from [EUR15]).

OAI can be used along with standard RF laboratory equipment (e.g. USRPs) to implement these functions to a sufficient degree to allow for real-time interoperation with commercial devices. It can also run in emulation/simulation mode, but it is out of the scope of this dissertation so it won't be discussed.

The UEs used with the platform can either be commercial UEs (smartphones/dongles) or the software radio OAI UE.

The software is freely distributed by the Alliance under the terms stipulated by the OSA license model, in Figure 3.2. For distribution of software and documentation, two licenses are used: OAI Public License V1.1 for the eNB and UE RAN and Apache V2.0 License for the EPC. To accept contributions from individuals and corporations, an Individual or Corporate Contributor License Agreement needs to be signed [Ope17a].

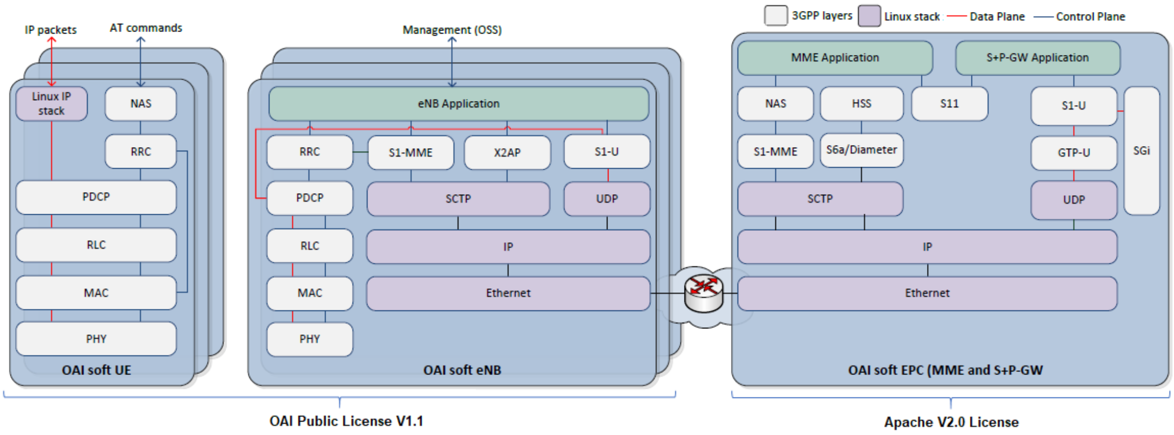


Figure 3.2: OpenAirInterface (OAI) Software Alliance (OSA) software stack and licensing model (adapted from [Ope17a]).

Some industrial users have already been working on OAI-based systems integrated with commercially-deployable remote radio-head equipment, as well as several universities and research institutes, including Instituto de Telecomunicações/Universidade de Aveiro with the Flexicell project [Fle15], an OAI-based Cloud Radio Access Network.

The LTE protocol stack implemented by OAI is shown in Figure 3.2. It provides the following PHY layer features [Ope17a, EUR17b]:

- LTE Release 8.6 compliant, with a subset of Release 10.
- FDD and TDD configurations in 5, 10 and 20 MHz bandwidth.
- Transmission mode: 1 (Single Input Single Output (SISO)) and 2, 4, 5 and 6 (Multiple Input Multiple Output (MIMO) 2x2).
- CQI/PMI reporting.
- All DL channels and signals are supported: PSS, SSS, PBCH, PCFICH, PHICH, PDCCH, PDSCH.
- All UL channels and signals are supported: PRACH, PUSCH, PUCCH, SRS, DRS.
- Hybrid Automatic Repeat reQuest (HARQ) support (UL and DL).
- Highly optimized base band processing (including turbo decoder).

Regarding the E-UTRAN protocol stack, the following features are provided [Ope17a, EUR17b]:

- LTE Release 8.6 compliant, with a subset of Release 10.
- Implementation of the MAC, RLC, PDCP and RRC layers.
- Protocol service for all Release 8 Channels and Release 10 eMBMS.
- Channel-aware proportional fair scheduling.

- DCI generation, UL power control and power headroom reporting (PHR).
- Integrity check and encryption using the *AES* and *Snow 3G* algorithms.
- Support of RRC measurement with measurement gap, but handover is still under integration.
- Standard S1AP and GTP-U interfaces to the core network.
- IPv4 and IPv6 support.

The EPC features are [Ope17a, EUR17b]:

- MME, S-GW, P-GW and HSS implementations. GTPv1-U and GTPv2-C application protocols are reused from the open-source software implementation of EPC called *nwEPC* [nwE12].
- NAS integrity and encryption using the *AES* and *Snow 3G* algorithms.
- UE procedures handling: attach, authentication, service access, radio bearer establishment.
- Transparent access to the IP network (no external S-GW nor P-GW are needed). Configurable APN, IP range and Domain Name System (DNS).
- IPv4 and IPv6 support.

OAI was introduced and its main features were presented. In the next subsection, the software platform will be described, with emphasis on its structure.

3.3.1 Software platform and coding structure

The OAI software is obtained from EURECOM's GitLab server [EUR17b], which is also where most of the information regarding the platform is, i.e. tutorials and frequently asked questions. It is divided in several branches, with `master` and `develop` being the most used. The `master` branch contains a stable version of the code, usually a few months old, while the `develop` branch contains a weekly updated version of the code, proving to be unstable at times. To retrieve the source code, a Git client is required. The installation process will be fully described in Appendix C.

In order to be able to contribute and commit code to the Git repository, one has to be added as a developer first. Then, if the defined coding guidelines were respected, a merge request can be submitted. Each Friday, a temporary integration branch is created, where all the potential commits are merged. After running a series of tests, they will either be accepted and pushed into the `develop` branch, or rejected and excluded from the integration branch.

The community is very active. The way for everyone to communicate is through mailing lists. The questions are visible to all subscribers, which is great for problem solving and solution sharing. Considering that the information regarding OAI is scattered all over its websites, or sometimes not updated or existent at all, it is of the utmost importance to be subscribed to the mailing list.

Because of the licensing model, the OAI source code is divided in two different repositories, `openairinterface5g` (eNB/UE RAN) and `openair-cn` (EPC), detailed in the following sections 3.3.1.1 and 3.3.1.2.

3.3.1.1 openairinterface5g

The source code for the eNB and UE radio access networks can be found in `openairinterface5g` and the functional units it implements are shown in Figure 3.3. Initially, the UE belonged to a different project, `openair1B`, which aimed to build a OAI UE with basic functionalities (fully integrated and stable with OAI eNB and EPC). It was merged with the `openairinterface5g` after its completion, in December 2016.

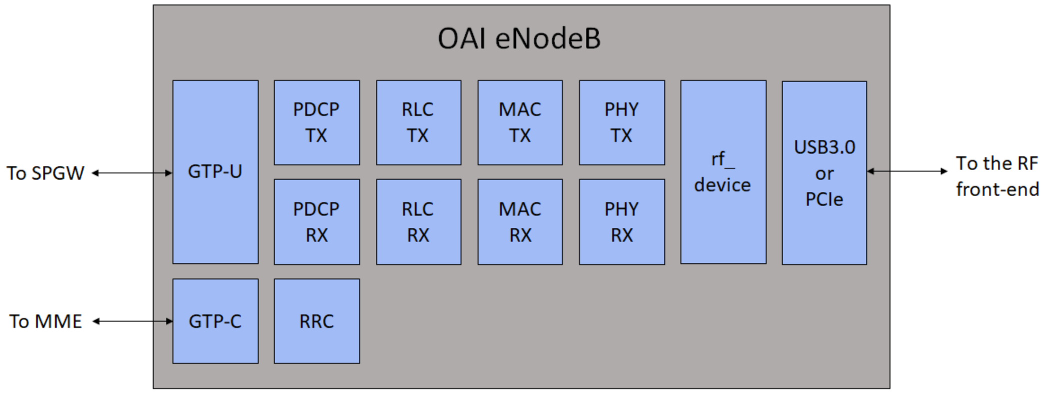


Figure 3.3: Functional units implemented by `openairinterface5g`

The structure of the source code is organized as follows [EUR17b]:

- cmake_targets: Build utilities to compile (simulation, emulation and real-time platforms) and generated build files. It contains the "mother" build_oai script, used to build the eNB and UE softmodems.

- targets: eNB and UE top-level wrapper for use with/without hardware in emulated and real-time modes. It is also where the hardware specific code is located (drivers, tools, etc.).

- openair1: Basic DSP routines for implementing a subset of Release 12 LTE specifications. It contains the PHY layer and PHY abstraction software.

- openair2: RLC, MAC, PDCP, RRC and X2AP implementations.

- openair3: S1AP, NAS and GTPV1-U, for both eNB and UE.

- common/utils: Some common utilities, such as the InTer-Task Interface (ITTI).

The E-UTRAN is configurable for a wide array of parameters according to the user's needs, like the E-UTRA band of the transceiver, its downlink/uplink frequency, number of resource blocks, number of antennas, duplex modes, power control and so on. For the eNB, this is achievable through the use of configuration files, available in Appendix C. The OAI UE doesn't have a configuration file per se, meticulous modifications have to be made in several files. The whole process is also explained in Appendix C, a guide on how to install and run the OAI platform detailing all the steps and changes required for the platform to work.

eNB configuration The parameter configuration file for the eNB (in Appendix C) is divided in six main sections:

- **Main parameters:** Configuration of the base station identity, TAC, MCC and MNC.
- **PHY parameters:** Configuration of the physical layer parameters - E-UTRA band, downlink/uplink frequency, power control, number of TX/RX antennas, TX/RX gain, hopping, frame type and so on.
- **Special Radio Bearer (SRB) parameters:** Configuration of special radio bearers parameters - poll retransmission timer, reordering timer, etc..
- **MME parameters:** Configuration of MME parameters - IPv4/IPv6 addressing.
- **Network interfaces:** Configuration of network interfaces - eNB S1-U IPv4 address, eNB S1-MME IPv4 address and the interface names.
- **Log configuration:** Choosing logger's level and verbosity on the terminal by taking into account all the layers and components of the network - hardware (HW), PHY, MAC, RLC, PDCP and RRC.

Only one configuration file can be used when running the eNB wrapper. After configuration, the eNB is ready to be built and executed. The UE configuration will be described next.

UE configuration The parameter configurations for the OAI UE (in Appendix C) are scattered through several files, which all come together in the building process. The two most important files are related to the NAS protocol. One of them defines all the Public Land Mobile Network (PLMN) network operators the UE will be able to recognize. The other is an USIM card data generator, which allows the user to define the OAI UE's simulated user and USIM card parameters, such as the IMEI, K and OPc, as well as a selection of known PLMNs from the previously described file.

Some other parameters are fed to the UE when executing the wrapper on a Linux terminal. These include the frequency band it's going to operate in, duplex mode (TDD or FDD), number of resource blocks, transmission and reception gain and so forth.

This completes the E-UTRAN software implementation description. The EPC will be described next.

3.3.1.2 openair-cn

`openair-cn` provides the source code for the core network entities S-GW, P-GW, HSS and MME. S-GW and P-GW are actually combined in a single entity, commonly named SPGW. Each entity runs as their own process and each procedure or protocol is run as a thread in these processes [Mar16].

The communication in MME and SPGW is made by means of ITTI, an intra-process communication system for message passing [Flo17]. Each interface adapter or protocol instance is assigned its own task and each task is woken up by events (messages, timers). HSS uses a threading architecture provided by the Diameter library, instead.

Each EPC network entity and its configuration in OAI will be discussed in the following sections.

HSS configuration HSS contains a database and a S6a thread. Prior to building the HSS, MySQL needs to be installed, so the network subscribers information can be programmed into a database, including the IMSI, IMEI, Mobile Station International Subscriber Directory Number (MSISDN), K, OPc and APN. This represents the AuC. Any user that tries to register to the network and whose information is not in the database will automatically be rejected.

freeDiameter [fre11] is also installed. It is an open-source Diameter implementation and it is used for S6a signalling.

Two main configuration files are required, one for MySQL and the other for freeDiameter. The first file is used to configure the MySQL server address, user name, password chosen during the installation process and name of the database where the network subscribers' information will be stored. The second provides the parameters needed for the Diameter protocol, such as the Fully Qualified Domain Name (FQDN) and the location of some required TLS[1]certificates.

MME configuration Just like HSS, MME also runs a S6a thread that uses freeDiameter.

The main configuration file is divided in two sections, MME and S-GW. As the P-GW and S-GW selections are not implemented yet, a mechanism is needed to replace it. Currently, only one PDN, one P-GW and one S-GW are supported. So, a S-GW IPv4 address for the S11 interface is assigned directly in the configuration file.

MME's main section allows for customization of the Diameter realm of the MME, the maximum number of operating eNBs or UEs, if the Emergency Attach is supported, etc.. Other sections include choosing the message queue size for ITTI, the S1AP outcome timer, the Globally Unique MME Identifier (GUMMEI) and TAI configuration parameters, the NAS integrity and ciphering algorithms preference order, the network interfaces for S1-C and S11 and logging configuration.

SPGW configuration As the S-GW and P-GW nodes are combined, the interfaces S5/S8 supposed to connect the two aren't used. SPGW makes use of the GTP module provided by Osmocom.

The SPGW configuration file is divided into S-GW and P-GW. The first allows for configuration of the network interfaces for S11 and S1-U, ITTI message queue size and logging options. The latter allows for customization of DNS addresses, SGi network interface, a pool of IP addresses available for UEs and if the outgoing UE traffic should be masqueraded - Source Network Address Translation (SNAT).

Now that the software platform was described, the hardware requirements to run OAI will be discussed.

3.3.2 Hardware requirements

OAI was designed to run in Intel x86 and ARM architectures. Currently, the ARM implementation is neither updated nor functional; so, an Intel GPP has to be used. The OAI eNB and UE require quite a lot of computational power, due to highly optimized DSP routines which make use of integer Single Instruction, Multiple Data (SIMD) instructions,

[1] Transport Layer Security

FFTs, turbo decoder and multi-thread parallel processing. The minimum requirements to run the software, according to OAI, are the processor families Intel i5, i7, Xeon and Atom, with a clock rate above 3 GHz and a minimum of 4 cores. The EPC, however, should run on any machine, or even in a virtual machine.

Regarding the operating system, the eNodeB and UE work on both Ubuntu 14.04 LTS 64-bit (`master` branch) and Ubuntu 16.04 LTS 64-bit (`develop` branch) Linux distributions. To ensure real-time operation, a low-latency kernel is required (version 3.19 for `master` branch and 4.8 for `develop`), along with the removal of all power management tools in the BIOS. This includes disabling Intel Enhanced SpeedStep, CPU frequency scaling, hyper-threading, Turbo mode, C-states and P-states.

The EPC also works on both Ubuntu 14.04 LTS and Ubuntu 16.04 LTS 64-bit. As its operation is not as time-sensitive as the eNB/UE's, the low-latency kernels and power management configurations are not required. However, the GTP module for SPGW demands at least a 4.7.7 Linux kernel, provided by OAI in an already pre-compiled Debian package.

OAI can be interfaced with several commercial SDR RF platforms. The currently supported hardware platforms are EURECOM's ExpressMIMO2 [Exp17], USRP B2X0/X310 [Ett17], bladeRF [Nua17] and LimeSDR [Lim17]. ExpressMIMO2 and USRP X310 acquire real-time data to/from the PC through PCIe, which is very efficient, as the memory is accessed directly (DMA). All the other options acquire data through USB 3.0, spending some extra CPU time for (de-)packetization of signals. In section 5.2.2, these platforms will be discussed and compared.

These two initial chapters aimed to provide the necessary knowledge required for the development and understanding of this dissertation's work. In Chapter 2, the LTE standard was detailed, specifically regarding the network architecture, protocols and physical layer aspects, such as the multiple access schemes, modulation, duplex modes and physical channels. Some of the LTE frequency bands were also introduced.

In Chapter 3, several software-defined radio experimentation frameworks that implement the LTE standard were analysed, with the purpose of picking the most adequate for the proposed work. EURECOM's OpenAirInterface (OAI) was chosen, as it is the most complete framework, implementing a basic User Equipment (UE), an evolved NodeB (eNB) and an Evolved Packet Core (EPC). The framework's software and hardware structure was detailed, describing the source code's division into access and core network, along with the configuration files that allow the definition of inputs for each network node, according to the user's needs. On the access network's side, the framework must be interfaced with a software-defined radio RF front-end, which will be discussed later in Chapter 5.

The following chapter will assess the possible non-conventional LTE deployment scenarios that could make use of a flexible software-defined radio LTE UE, implemented using the chosen framework.

Chapter 4

Non-Conventional LTE Deployment Scenarios

In this chapter, a description of the deployment possibilities using OpenAirInterface (OAI)'s framework is presented, in section 4.2, followed by a listing of non-conventional LTE deployment scenarios that would benefit from a flexible and compact software-defined radio UE implementation, in section 4.3.

4.1 Introduction

The currently available cellular networks offer a wide array of telecommunication services that contributed deeply in the way we communicate, work and live. Their existence and availability is important in such a way that mobile telecommunication operators continue making a great numbers of investments, so the adequate level of coverage and QoS is guaranteed to their clients. However, there are still exceptions. Sometimes, the installed networks fail in a generalized manner and/or can't provide the adequate coverage and support. In these situations, a flexible solution that fully implements an LTE network or aids the already existing ones is required.

The chosen open-source LTE framework from the previous chapter, OpenAirInterface (OAI), has a high degree of flexibility and configurability, allowing the implementation of non-conventional scenarios. These will be described later, in section 4.3. Several deployment possibilities for the OAI platform are available. The most relevant ones are described in the next section.

4.2 OAI Deployment possibilities

The OAI platform can be deployed in different scenarios, involving commercial components or not. For transparency, the various network nodes implemented by the framework will only be referred to by their name (e.g. "UE" for the user equipment, instead of "OAI UE"). The working combinations, so far, are the following:

- Commercial UE ↔ eNB ↔ Commercial EPC
- Commercial UE ↔ Commercial eNB ↔ EPC

- Commercial UE ↔ eNB ↔ EPC
- UE ↔ eNB ↔ EPC
- UE ↔ eNB

The UE's interoperability with commercial eNBs or EPCs was not successfully achieved yet, it is still in experimental mode. Only the last three deployments are interesting for this dissertation's work, as using a commercial eNB or EPC would not allow the total reconfigurability required. These will be described in the following sections.

4.2.1 UE connected with the eNB

In this scenario, there is no core network, which means that there isn't access to the Internet. The S1AP and GPRS Tunelling Protocol (GTP) protocols are bypassed, meaning the eNB and the UE communicate through a radio link. The IP packets are exchanged with the upper layer through an network device driver created by the framework, instead of through the regular NAS protocol.

Figure 4.1 represents this scenario. The radio components are represented by a black box. This is because several combinations can be used.

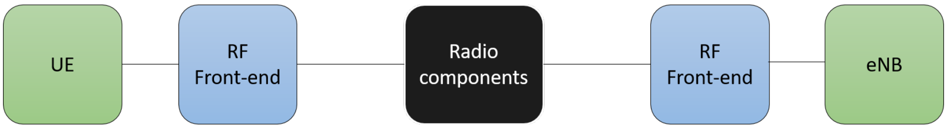


Figure 4.1: OAI deployment scenario: UE connected with the eNB.

As this scenario does not provide network access, it is not relevant for this chapter. It is used for debugging purposes only, during the implementation of the other deployments. It rules out problems such as: checking if the UE connects to the eNB, to figure out if there are problems with the core network or vice-versa and to check the radio link configuration between the two.

4.2.2 Commercial UE or UE connected with the eNB and EPC

Two different small-cell scenarios are represented in this section. Their main difference is in the user equipment: one is a commercial equipment, such as a smartphone or a dongle (in Figure 4.2), while the other is created by the used LTE framework and run in a general-purpose processor (in Figure 4.3). The eNB is built with the S1AP and GTP protocols and is connected with the Evolved Packet Core (EPC), providing connectivity to the Internet.

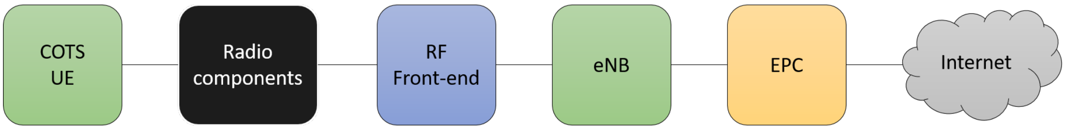


Figure 4.2: OAI deployment scenario: Commercial off-the-shelf (COTS) UE connected with the eNB and EPC.

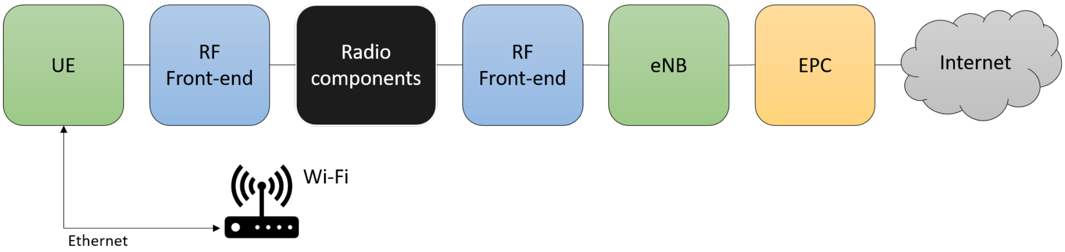


Figure 4.3: OAI deployment scenario: UE connected with the eNB and EPC.

The connection between the eNB and the EPC is made with an Ethernet cable, while the connection between the eNB and the UE is assured with a radio link. Regarding the radio components' black box in both figures, several configurations are possible, depending on parameters like the duplex type used for transmission.

The commercial UE deployment scenario is the most stable and optimized OAI implementation. It is used as a testing reference for the results to be achieved with the UE deployment. The commercial UE is not relevant for the purpose of this chapter, as it is not flexible or easy to reconfigure. For example, a smartphone is already bought with a certain number of known frequency bands and these are the ones it will recognize forever.

The last deployment, and the one that is relevant for this chapter due to its flexibility, is the UE implementation running in a General Purpose Processor (GPP). After being connected to the LTE network, it can share its mobile data with nearby devices by creating a Wi-Fi hotspot, using an Access Point (AP) or the computer's Wi-Fi module, as represented in Figure 4.3.

The following section will list some non-conventional LTE deployment scenarios that could make use of a compact, flexible and easily deployable software-radio UE, able to share its mobile data with surrounding users through a Wi-Fi hotspot.

4.3 Non-Conventional Scenarios

For most of the non-conventional scenarios that will be described in this section, it would be extremely useful if the user could just authenticate to the temporary network by using his commercial UE without needing to establish a connection with its home network first, in order to verify the USIM card credentials. However and after much research, it was confirmed that such a scenario is not possible to deploy (atleast according to the 3GPP's specifications for LTE). It would only work if the temporary network's database already contained the user's USIM card information. This is because of LTE's mutual authentication procedure, described in Appendix A, which basically consists on a comparison of keys generated by the UE and the network - if both groups of keys aren't exactly the same, the procedure terminates -, followed by encryption and integrity protection algorithms.

There's a specific type of attachment, named Emergency Attach, which authorizes the UE to attach to the network without authentication, but it is only allowed for IMS emergency calls. This is also explained in Appendix A. Any other functionalities are blocked, unless the mutual authentication procedure is completed.

Nevertheless, a workaround is possible and a connection to the LTE network can still be provided to nearby users, as is described in the next section.

4.3.1 Sharing the LTE network via Wi-Fi hotspot

A software-defined radio implementation of the UE, interfaced with an RF front-end platform, is attached to the created LTE network with access to the Internet. The UE is now able to share its LTE connection with the surrounding devices, either by Ethernet or a Wi-Fi hotspot, as in Figure 4.4. This way, only one UE is required to complete the mutual authentication procedure, while providing free Internet coverage to a specific location via Wi-Fi.

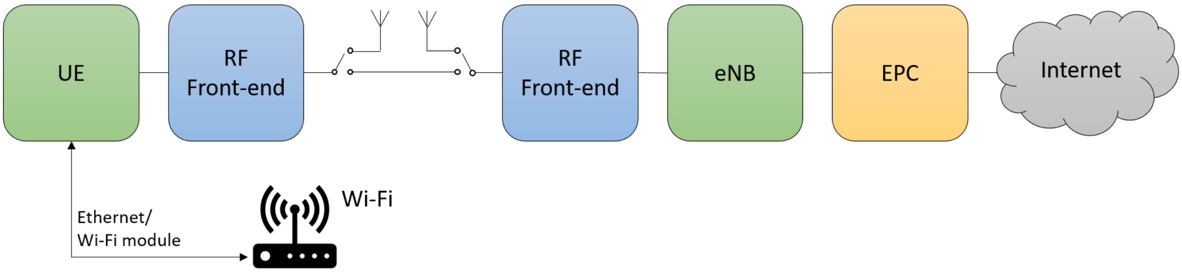


Figure 4.4: Non-conventional LTE deployment scenarios architecture.

All of the subsequent scenarios will use this network sharing method.

4.3.2 Catastrophe

It is common that in catastrophe scenarios, e.g. hurricanes and wildfires, damage is inflicted to the existing telecommunications systems (as in Figure 4.5) and the coverage must be re-established as soon as possible, either for emergency purposes or ordinary communications. The same can be said for war scenarios, where radio towers might be destroyed, for example, and communications become unavailable. The existence of an easily deployable UE that shares a free Internet connection in a certain area via a Wi-Fi hotspot would be very useful.

Figure 4.5: Destroyed communications tower, caused by a wildfire.

A possible implementation architecture is shown in Figure 4.4. In terms of radio configurations, the connection between the UE and the base station can be cabled, avoiding any air interface losses but requiring them to be close to each other, or wireless, using antennas.

4.3.3 Remote locations

There are still locations on the globe where no regular mobile network operator coverage is available. These can be third world countries, on board of cruise ships or just sporadic events in remote places. A temporary or even seasonal existence of a LTE mobile network that provides Internet access would be very interesting from an economic point of view.

A possible implementation architecture is shown in Figure 4.4. For remote or rural locations, the utilization of FDD frequency band 31 should be explored. It is the lowest LTE frequency band on the spectrum, 450 MHz, which means that the coverage it provides is bigger than any other bands. The only disadvantage is that it only allows bandwidths up to 5 MHz. Band 31 is in the same frequency range as one of the Terrestrial Trunked Radio (TETRA) bands in Europe, used for emergency services.

4.3.4 Opportunistic use of TDD in frequency bands licensed for different purposes

Certain frequency bands are licensed for different purposes than LTE transmission. This is the case of TDD band 40. Such a premise should be taken advantage of, so the transmission on the TDD band 40 and similar frequency bands should be tested. Once again, the architecture of the system could be the one in Figure 4.4.

4.3.5 Unlicensed spectrum and custom frequency bands

LTE-Unlicensed (LTE-U) was created by cellular network operators in order to find a solution for the increasing demand in mobile data. The idea is that it offloads data traffic by accessing the unlicensed 5 GHz frequency band [LMRZ17].

Considering the flexibility of the UE implementation, LTE in the unlicensed spectrum or any other custom frequency bands are a possible and a non-conventional scenario that could be implemented. The proposed architecture is the one in Figure 4.4, with either radio configurations.

The two last proposed scenarios are the most conventional amongst the group.

4.3.6 Investigation infrastructures

For a different scenario, there could also be investigation infrastructures with the need for open-source and modular platforms that grant an elevated level of configurability and observability of the LTE network system (e.g. university campuses with platforms for demonstration and validation of new systems and telecommunication services).

One of the possible architectures would be the same as in Figure 4.4, but considering it is a scenario for investigation purposes, it will be subject to change.

4.3.7 Network reinforcement

This is a very conventional scenario, when compared to the previous ones. A reinforcement of network coverage and signal might be required. Temporary installations for event support, such as conferences (in Figure 4.6), gatherings, festivals and sport events, where the number of users on the network is greater than usual and the existing deployments might not be enough are one example. Other possibilities are hotel rooms, where hotel owners might be interested in reinforcing the network coverage of their establishments.

Figure 4.6: Conferences, where a signal reinforcement might be required.

The proposed architecture for this scenario is also the one in Figure 4.4, using a cabled setup in between the UE and eNB and then sharing the network through the Wi-Fi hotspot. A cabled setup is preferred, as the air interface in such locations is surely overcrowded.

In this chapter, the framework-related deployments were presented, being that only one of them was relevant to this chapter, the software-defined radio UE implemented in a General Purpose Processor (GPP). Then, the network architecture to be used in all of the presented scenarios was defined, with differences only on the radio setup level. To finalize, several non-conventional LTE deployment scenarios were discussed.

In the following chapter, some of these scenarios will be implemented and explained in a more detailed way, along with the used network configurations and parameter changes.

Chapter 5

Implementation

In this chapter, the implementation of the setup is described. It begins with an introduction showing the block diagram of the network components and their radio configurations, in section 5.1. Then, each network component is described: the UE in section 5.2, the eNB in section 5.3 and the EPC in section 5.4. The subsequent section 5.5 contains information on how the obtained LTE mobile data is shared by creating a free Wi-Fi hotspot.

5.1 Introduction

As was introduced in Chapter 4, various scenarios can be deployed with OAI. This dissertation intends on deploying a fully reconfigurable LTE user equipment that communicates with the access and core networks and has Internet access.

Figure 5.1 represents the full implementation setup, merging all of the used configurations. Two types of UE implementations were deployed, each with different radio setups, described below:

- Commercial UE, represented by *(a)* in Figure 5.1. It is then connected to the network, which has one of the following radio setups:
  - two antennas, represented by the number *(1)*.
  - one antenna and one duplexer, represented by the number *(2)*.
- Software-defined radio UE implemented in a GPP, represented by *(b)* in Figure 5.1. It is first interfaced with a RF front-end, and then connected to the LTE network with one the following radio setups:
  - two duplexers and two antennas, one of each per side, represented by the number *(3)*.
  - four antennas, two on each side, represented by the number *(4)*.
  - two RF attenuators, one for each pair of ports, represented by the number *(5)*.
  - two duplexers and 1 attenuator in between, represented by the number *(6)*.

After the radio configurations are the base station's RF front-end, followed by the eNB's computer, which is connected with an Ethernet cable to the EPC's computer. The EPC is connected to the Internet with an Ethernet cable.

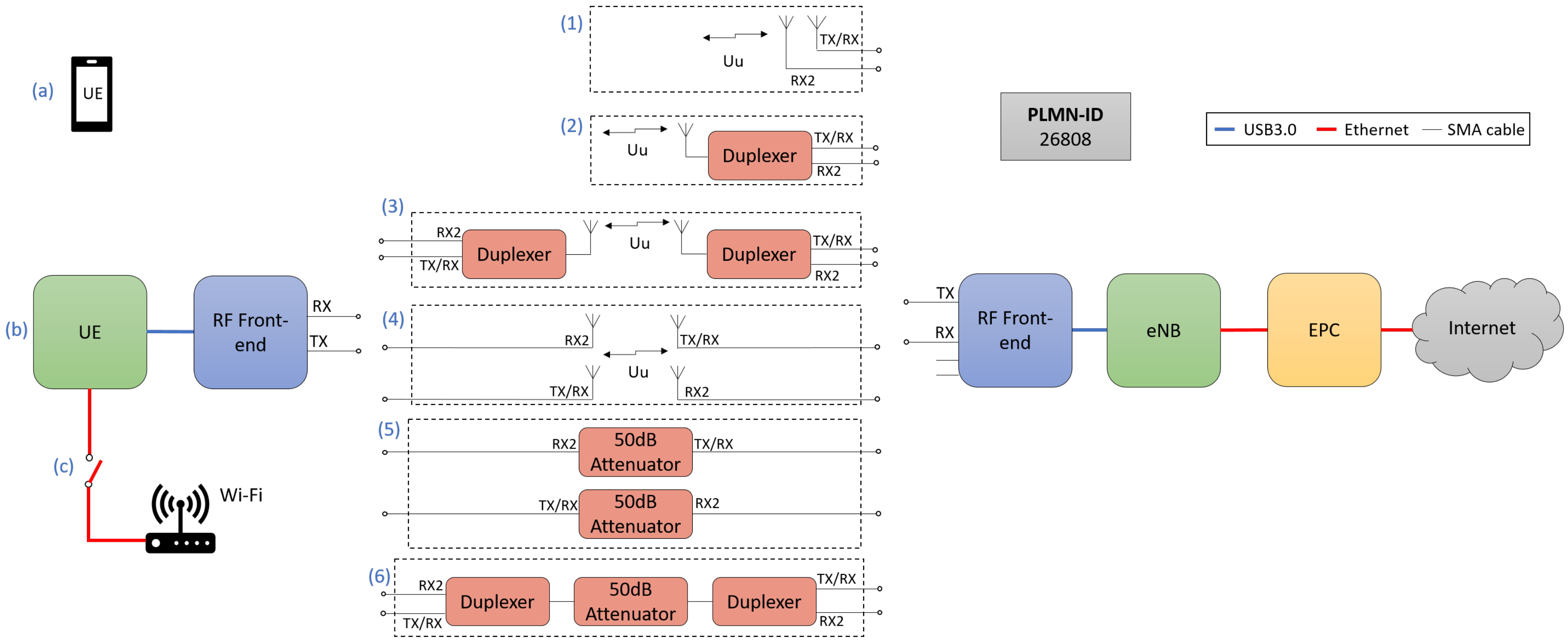


Figure 5.1: Implementation setup, including two different UEs, various radio configurations and the possibility to share mobile data through a Wi-Fi hotspot.

With the SDR UE implementation, the user can choose whether or not he wants to share its mobile data with the surrounding devices, by means of a Wi-Fi hotspot, which can be created with an Access Point (AP) or, if available, make use of the Wi-Fi module on the computer.

Each component's implementation will be described in the following sections.

5.2 UE Platform

In this section, the processing modules used for the UE implementation are described, followed by the used RF front-end and finishing with the used radio components.

5.2.1 Processing Modules

OAI's source code demands for some computationally heavy processing, mainly for the eNB and UE softmodems. First, the UE is implemented in a powerful mini-computer, the *Intel NUC*, and is then migrated into a more compact and energy efficient single-board computer, the *UP Squared.*

5.2.1.1 Intel NUC Mini-computer

The Intel NUC Kit NUC6i7KYK was chosen to run the UE implementation, with a quad-core Intel Core i7-6770HQ @ 2.60GHz and 16 GB of memory. This is one of Intel's mini-PCs.

It is running the 64-bit Linux Ubuntu 16.04.3 LTS operating system, with kernel version 4.8.0-46-lowlatency. This low-latency kernel is used to help with the real-time input from peripherals. A real-time kernel could be used as it provides the lowest latency possible, but it also increases the power consumption a lot.

Other power management features have to be disabled, most of them in the Basic Input/Output System (BIOS), described here:

- C-states and P-states. The P-states are voltage-frequency pairs that set the speed and power consumption of the coprocessor [Pow14]: the lower the voltage, the lower the frequency and power consumption. The system requires maximum performance, so the P-states are set to their maximum state, so there's no power-saving and, consequently, performance loss. C-states are idle power saving states. During a C-state the processor is idle, meaning it's not doing any processing and, therefore, saving power. The C0 state represents the "always on" non-idle state; all other states will be disabled.

- Central Processing Unit (CPU) frequency scaling enables the operating system to scale the CPU frequency up or down manually in order to save power, as well as automatically, depending on the system load. There are several power schemes for the CPU, named governors. The default is the `ondemand` governor, which scales the power with the system load, so it is switched with the `performance` governor, which runs the CPU at its maximum frequency [CPU17].

- The intel_powerclamp driver, related to C-states, is used for reducing power consumption and is also disabled.

- Other tools in the BIOS also have to be disabled, such as Intel's Hyper-Threading technology, CPU frequency control, Intel SpeedStep and Turbo modes.

The source code is pulled from GitLab's `openairinterface5g` and then switched to the `develop` branch (git tag 17b9a9e).

5.2.1.2 Single-Board Computer (SBCs)

It is of this dissertation's interest to transform the PC running the UE softmodem (software modem) into something more compact, energy-efficient and easier to carry/drop onto a chosen location, while still being able to run the computationally intensive OAI code.

Ideally, an ARM-based platform would be picked. The ARM architecture is based on Reduced Instruction Set Computers (RISC), opposed to the Complex Instruction Set Computers (CISC) processors from Intel. This is why ARM-based processors have low power consumption, while still maintaining a high performance. However, OAI is not optimized for ARM, as all digital signal processing functions use integer SIMD instructions, from Intel x86 architecture. Optimizing the entire code for ARM-based processors was too much work for one person, so this was not considered.

The best way to combine an Intel-based platform with low-power consumption and small dimensions is by using a Single-Board Computer (SBC). There are several options in the market, described in Table 5.1. These should all be able to run the UE softmodem in real-time, with the adequate modifications.

Table 5.1: Comparison of the commercially available SBCs.

	UP Squared	**UDOO x86 Ultra**	**Congatec Conga PA5**	**Commel LP-173J**	**Intel Joule 570x Dev Kit**
CPU	Intel Pentium N4200 @ 2.5 GHz	Intel Pentium N3710 @ 2.56 GHz	Intel Atom x7-E3950 @ 2 GHz	Intel Celeron J1900 @ 2.42 GHz	Intel Atom T5700 @ 2.4 GHz
RAM	8 GB	8 GB	8 GB	8 GB	4 GB
eMMC	64 GB	32 GB	-	-	16 GB
Power Supply	+5V, 6A	+12V, 3A	+12V	+6∼27V	+12V
Additional Info	4x USB 3.0 2x Gb LAN	3x USB 3.0 1x Gb LAN WiFi	3x USB 3.0 2x Gb LAN	1x USB 3.0 1x Gb LAN	2x USB 3.0 WiFi
Price	290€	320€	340€	N/A	340€

As can be observed in Table 5.1, both Congatec's *Conga-PA5* [con17] and Commel's *LP-173J* [Com17] have no embedded Multi-Media Controller (eMMC) memory, which would require the attachment of an extra mSATA SSD[1]. Not only it would be more expensive, it would also increase the platform's size. Therefore, these two SBCs were discarded. *Intel Joule 570x Development Kit* [Int17] is more expensive and has worse attributes than the UP Squared [UP 17b] or the *UDOO x86 Ultra* [UDO17]. Thus, it was also discarded.

There are now two platforms left. The *UP Squared* is cheaper and although it might draw up to 6A of current, it still comes under UDOO's power consumption. Additionally, it has a slightly better processor and a bigger eMMC and the community is quite active, which is

[1] mini-Serial ATA Solid State Drive (mSATA SSD)

good for support in case anything goes wrong. The only advantage of the UDOO board is the Wi-Fi module, allowing for easier sharing of the network with other users, without the need for an external AP or Wi-Fi dongle. It's not a good enough reason to choose it over the *UP Squared*, so it was also discarded.

Now that the SBC is chosen(*UP Squared*, a quad-core Intel Pentium N4200 @ 2.5GHz, with 8 GB of memory, in Figure 5.2), it needs to be prepared to run the UE's source code. The process is different from the one described for the normal computer. First of all, an active fan is installed on the board, instead of the passive heatsink that was already pre-installed. This allows for better refrigeration of the processor and other temperature-sensitive components.

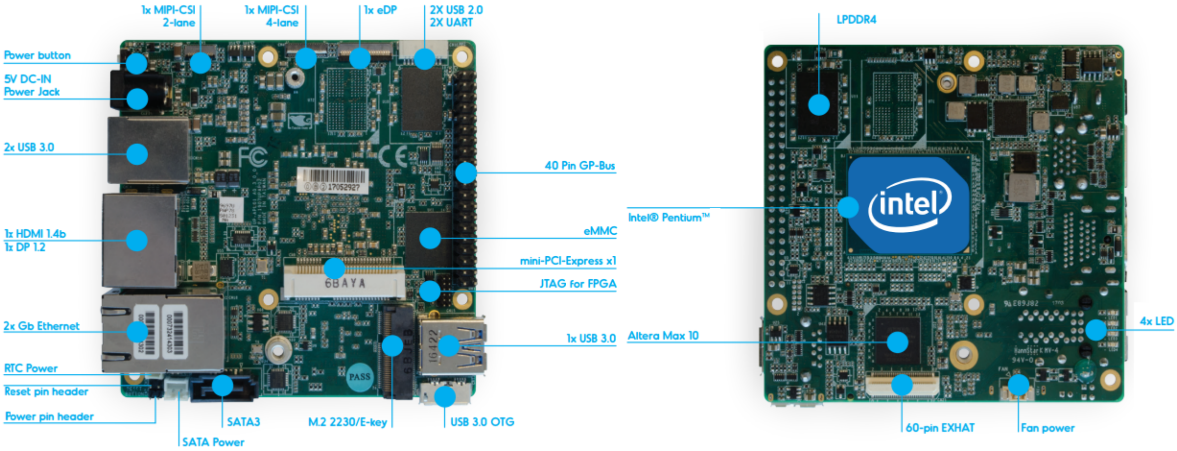


Figure 5.2: Chosen Intel-based SBC, the *UP Squared* (taken from [Up 17a]).

Afterwards, for the software preparation, 64-bit Linux Ubuntu 16.04.3 LTS was installed, with a 4.8.0-46-lowlatency kernel, just like on the previous implementation. In order to have real-time communication, the power management features should be disabled. In this case, that won't be possible; if the turbo boost is disabled from the BIOS, the maximum achievable CPU frequency will be 1 GHz. This is obviously too low, considering the recommended speed on OAI's website is 3.3 GHz. Thus, another approach has to be taken.

The UE is divided in three main threads, meaning that each individual thread can be mapped into a specific CPU core, leaving one free core for non real-time Linux operations. This line of processors isn't equipped with Intel's hyper-threading technology, so there won't be any problems with virtual cores. The core mapping goes like this:

- Core 0, being the default Linux core, is the one left for non-real time system processing.
- Core 1 and core 2 will be assigned the processing of even and odd subframes, one thread for each.
- Finally, core 3 will be responsible for OAI's internal scheduling and for I/Q samples acquisition.

OAI is then compiled without the deadline scheduler it usually uses, as in this configuration it will create errors with the front-end's driver later on; a first-in, first-out (FIFO) scheduler is used instead. Using Linux's CPU management utility `cset` [Cpu13], one can set up a `shield` on three CPUs, leaving the unshielded CPU running all system current tasks and also new ones that are spawned. The user is now free to choose which processes/tasks he wants to run in the shielded CPUs (terminal commands are in the C Appendix). Additionally and in order to improve the hardware latency, the front-end's Interrupt Request (IRQ)s can be moved to the same core that runs the I/Q management, core 3.

The *UP Squared* is almost OAI-ready. The cores must be forced to run at their maximum frequency, so the CPU's scaling governor has to be changed to `performance`. In order to not waste any processing capabilities, all peripherals except the RF front-end are removed (mouse and keyboard), as well as the monitor. The board will now be controlled remotely through an Secure Shell (SSH) session, which allows for remote login using a different computer.

5.2.2 RF Front-end

The computers running the eNB and UE must be connected to a software radio front-end. The commercially available hardware platforms have different characteristics, ranging from the form of data acquisition, version of the FPGA and RF chip to the frequency range, available bandwidth for the transceiver and output power. The platforms supported by OAI were analysed and their characteristics are shown in 5.2.

Table 5.2: Comparison between the SDR platforms supported by OAI.

	ExpressMIMO 2	**LimeSDR**	**bladeRF x40**	**USRP B210**	**USRP X310**
Data acquisition	PCIe	USB 3.0	USB 3.0	USB 3.0	PCIe, Gbit LAN
RF bandwidth	28 MHz	56 MHz	28 MHz	56 MHz	120 MHz
Duplexing	FDD/TDD	FDD/TDD	FDD	FDD/TDD	FDD/TDD
Frequency range	250 MHz - 3.8GHz	100 KHz - 3.8GHz	300 MHz - 3.8GHz	70 MHz - 6 GHz	DC - 6 GHZ
RF chip	LMS6002D (x4)	LMS7002M	LMS6002D	AD9361	N/A
Output power	10 dBm	10 dBm	6 dBm	10 dBm	10 dBm
Price	3320€	260€	360€	1250€	5500€

From the selected platforms, the ones to be discarded first are the ones acquiring data from PCIexpress (PCIe). *ExpressMIMO2* [Exp17] was developed by EURECOM [EUR17a] specifically as a front-end for OAI, but the most recent versions of the code are not optimized for the platform. PCIe is also not as convenient as Universal Serial Bus (USB) when it comes to portable solutions. Along with *USRP X310*, they're also the most expensive boards.

bladeRF x40 [Nua17] does not support TDD mode, so it is not useful for this dissertation. The remaining platforms are the *LimeSDR* [Lim17] and *USRP B210. LimeSDR* is a low-cost solution and their characteristics are similar. However, there were already some USRPs in the laboratory, and the USRP is the most stable and most tested platform by the OAI developers.

Therefore, the *USRP B210* (Figure 5.3) is chosen as the SDR front-end for the setup's eNB. It is bus-powered and connected to the computer with through an USB 3.0 SuperSpeed port. Trying to connect it to an USB 2.0 port will cause a massive loss of samples and real-time issues, as it is just not fast enough. For this reason, and even though it wasn't mentioned in section 5.2.1.2, the need for atleast one USB 3.0 port also went into the decision process for the UE's single-board computer.

USRP's *B200mini* (Figure 5.3) is also used in the setup. There are two main differences between the two: the *USRP B210* allows for MIMO (2x2) operation and the *B200mini* SISO (1x1), and the *B200mini* is the size of a business card, so it is used on the UE side. The *B200mini* is also cheaper, costing about 800€.

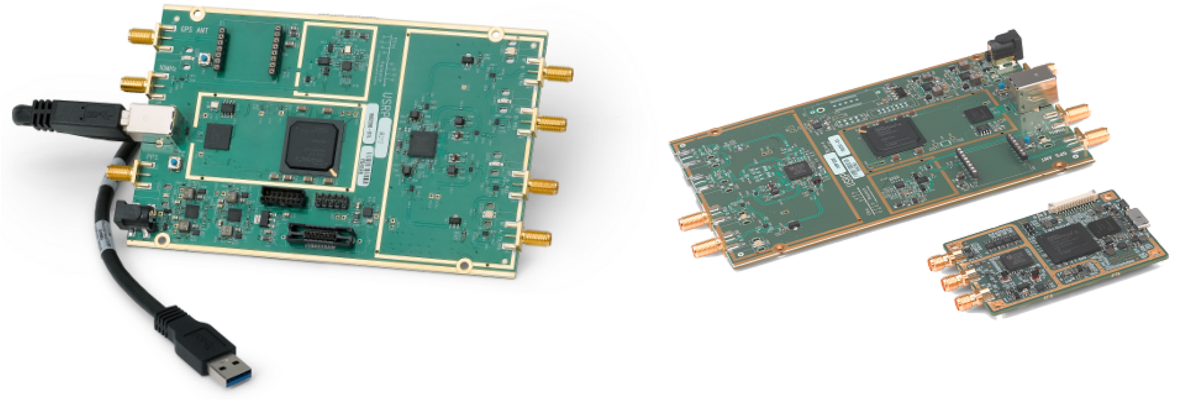


Figure 5.3: Used RF front-end platforms. On the left, *USRP B210*. On the right, a comparison between the USRP B200 and B200mini (adapted from [Ett17]).

OAI supports the Ettus USRP B-series products via the USRP Hardware Driver (UHD) driver, included in the building process of the softmodems.

5.2.3 Duplexers and additional RF equipment

Additional RF equipment must be connected to USRP's TX and RX ports, depending on the test setup: passive SAW duplexers, SubMiniature version A (SMA) fixed attenuators, Delock's LTE omni-directional SMA antennas (1-4 dBi) [Del17] and SMA coaxial cables.

The number of radio services has increased a lot over the last decade. For one service, the radiated signal of another service is an interferer. Therefore, frequency filters are necessary for all reliable radio services. OAI's over-the-air communication is very prone to errors. Even though the source code is more optimized for FDD than TDD, trying to transmit samples with two antennas (one for each USRP's RX and TX port) is not an easy task. Most samples get dropped, causing the eNB to reject the UE and remove it from the network.

Surface Acoustic Wave (SAW) filters are frequency filters suited from a few MHz to 3 GHz, which protect the service from interferers and make sure that most of the wanted signal will be forwarded to the receiver antenna. When transmitting, SAW filters are used to suppress the radiation of undesired harmonics. When receiving, SAW filters improve the selectivity (attenuation outside the pass-band) of the front-end. The image frequencies are rejected and powerful out-of-band interfering signals are blocked [Qua09].

As an LTE frequency band is divided into downlink and uplink operating frequencies, with different center frequencies (FDD only), so two SAW filters are required, which can be merged into a SAW duplexer. A duplexer is a three-port filtering device which allows transmitters and receivers operating at different frequencies to share the same communication path, usually an antenna, as represented in Figure 5.4.

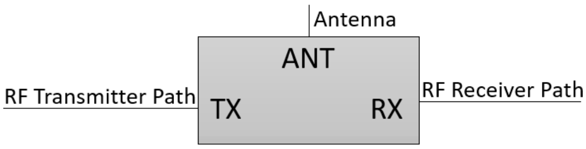


Figure 5.4: Basic duplexer representation.

In the context of this dissertation, passive SAW duplexers for three different LTE FDD bands were designed and built, in order to test different band operation, prove the flexibility of the device and improve the radio signal quality. SAW duplexer chips for LTE small-cell were used, for the frequency bands 1, 5 and 7. The duplexers' packages are very small - Surface-Mount Technology (SMT) - and their power-handling capabilities are not so great, meaning they only work for low-power transmission.

The influence of the Printed Circuit Board (PCB) layout gets stronger with higher frequencies. Too long lines or too thin lines between the SAW filter and the additional components will add additional losses [Qua10]. The external matching circuits recommended in the datasheet of each component are different for each bands. All informations regarding the layout, board design and a discussion of the obtained S-parameters are presented in Appendix B.

All developed SAW duplexers provide up to 10 MHz more usable passband than what is defined by 3GPP for the specific E-UTRA bands, in both downlink and uplink paths.

The duplexers for Band 1 and Band 7 work as expected, with the usable passbands 1920-1980 MHz (UL) / 2100-2170 MHz (DL) and 2490-2580 MHz (UL) / 2620-2710 MHz (DL), respectively. The measured signal attenuation of the two bands is flat and near the 0 dB mark through the whole passband. Both duplexers provide a good insulation with the surrounding frequencies by highly attenuating them, meaning their selectivity is high, as shown in Figures B.4 and B.5 of Appendix B.

Band 5's duplexer, however, has a lot of ripple in its usable passband - 830-855 MHz (UL) / 870-905 MHz (DL). Several problems may be causing this effect. As the components are very small and were hand soldered, this was thought to be the problem. However, after re-soldering, the ripple remained. It could also be a matching circuit problem, but according to the component's datasheet, no matching is required. The boards for Band 5 were the first to be designed and printed and the copper lines for the TX and RX ports are very thin for almost 1 mm between the duplexer's chip and the thicker copper wire, as can be seen in Figure B.2 of Appendix B. This could be causing the issue. There's still around 10 MHz of usable bandwidth for each UL and DL frequencies, so it will work for testing with 25 LTE resource blocks (5 MHz of bandwidth).

5.3 eNB Platform

The base station is where most signal processing occurs and strict real-time operation is required. If a number of samples is not processed in time, connections are dropped and the system shuts down. Therefore, the eNB will be ran on the best computer of the three components: a six-core Intel Core i7-6800K @ 3.40GHz, with 32 GB of memory.

It is running the 64-bit Linux Ubuntu 16.04.3 LTS operating system, with kernel version 4.8.0-46-lowlatency. This low-latency kernel is used to help with the real-time input from peripherals. A real-time kernel could be used as it provides the lowest latency possible, but it also increases the power consumption a lot. Other power management features have to be disabled, as described in the UE's section.

The RF front-end it is connected to, as was already described, is the *USRP B210*.

5.4 EPC Platform

The EPC code does not require heavy processing and it could even be ran in a virtualized environment. Therefore, the worst computer of the three was used for the core network: a quad-core Intel Core i5-661 @ 3.33 GHz, with 8 GB of memory. The used operating system was the 64-bit Linux Ubuntu 16.04.3 LTS, with kernel version 4.7.7-oaiepc, already pre-compiled with the GTP kernel module from Osmocom required for the SPGW.

There's also no need to remove the power management features from the BIOS, or the processor frequency scaling.

5.5 Sharing the LTE data through Wi-Fi

Before sharing the network, some additional routing has to be performed on the UE. Thanks to the SPGW's built-in SNAT functionality, the outgoing packets from the private LTE network are allowed to go to the Internet. When a commercial UE attaches to the network, it is given an IP from the IP address pool, configured with the network's DNS, and is then able to access the Internet by going through a default gateway which performs SNAT.

The UE, however, is not properly configured when it attaches to the network. After the attach procedure is complete, a virtual network interface named `oip1` is created and assigned an IP from the IP address pool. In this state, one can ping external IPs (e.g. Google DNS servers: 8.8.8.8) to test if the network is working, but it is not possible to browse the Internet. Some modifications are required.

First of all, a default IP route gateway should be assigned to the new network interface, `oip1`. Currently and unlike the commercial UEs, OAI UE is also not properly assigned with the network's DNS primary and secondary address. This can be solved by using a proxy, or by just adding the DNS configurations manually to the OAI UE's computer. As all configurations are known, the second option is chosen. After a network restart, the user should now be able to browse the Internet.

Now that the Internet is properly configured, the UE can run as is, but the ultimate goal of this dissertation is to have a device that provides Internet to multiple users. There are three options, discussed in the following subsection.

5.5.1 Sharing the network

The network can be shared with other users by using an Ethernet cable connected to a single computer or an AP/router, or by using the computer's Wi-Fi module to create an hotspot. The last two options were tested.

When running the UE on a computer, the network was shared through the built-in Wi-Fi module. Using Ubuntu's Network Manager GUI, it's possible to create a Wi-Fi hotspot that shares the computer's existing connection with others. If no password is configured, a free hotspot is now available.

When running the UE on a SBC, the network was shared by an Ethernet cable connected to the LAN port of an AP. The Ethernet port can be configured by manually routing and using SNAT, or with the Network Manager, by configuring the Ethernet port to share the computer's network. When configuring the AP, the Local Area Network (LAN) is configured as 10.42.0.2 and it is critical that the Dynamic Host Configuration Protocol (DHCP) functionality is disabled, otherwise it won't work. Again, if no password is configured, a free Wi-Fi hotspot is now available.

This chapter described the implementation of a reconfigurable LTE network that can be interfaced with a commercial UE or a software-defined radio UE for different radio configurations modes. After being successfully attached to the network, the UE will be able to share its LTE mobile connection with the surrounding devices, by creating a Wi-Fi hotspot.

The non-conventional scenarios described in Chapter 4 could all be deployed with the implementations described in this Chapter. For the catastrophe and remote location scenarios, the eNB could also be condensed into a smaller but powerful computer, like the *Intel NUC*, and the EPC ran in a virtualized environment. This way, this LTE implementation could be dropped anywhere it was required and provide access to the Internet.

For the other scenarios (opportunistic use of TDD bands, unlicensed spectrum and investigation infrastructures), the current implementations would suffice, at least for testing purposes.

In the next chapter, the test setups' results are presented.

Chapter 6

Results

This chapter begins with an introduction to the implemented setups and the measurements that will be performed, in section 6.1. Afterwards, the setup is described in section 6.2, along with the tools used for testing. Then, the test results are presented for the different implementations: section 6.3 for the commercial UEs, section 6.4 for the UE running on an *Intel NUC* mini-computer and section 6.5 for the UE running on the single-board computer *UP Squared*.

6.1 Introduction

Regarding the main purpose of this dissertation, which is the ability of having a compact, flexible and fully reconfigurable UE that works in a number of previously described scenarios, it is of great importance that the implemented setups can provide a reliable service to the user. If possible, the user should be able to browse the Internet, watch videos on Youtube, make Skype calls and so on.

Three major implementations were deployed. The only network element that varies with each implementation is the UE. First, a commercial UE is used, moving on to an UE running on Intel's NUC mini-computer and finalizing with it running in a single-board computer. The implementation with a commercial UE is the one that supposedly works best in OpenAirInterface (OAI), so it will be used as a reference on what results the others should achieve.

To evaluate the performance of each implementation, throughput and latency measurements were performed, along with the visualization of software oscilloscopes (softscopes) containing, for example, a representation of the received signal in dB or the constellations of different physical channel modulations. Additional entities are also taken into account and are reported by the UE to the eNB. These are the Power Headroom Report (PHR) and the Channel Quality Indicator (CQI), and will be described in the following section.

6.2 Setup and Tools

This section contains a description of the implemented setup, followed by the tools used to measure its performance.

6.2.1 Setup

In Figure 6.1 is the block diagram of the final setup, with the different UEs used for testing and their radio configurations, described in the previous chapter. The IP addresses assigned to each component are also presented in the Figure.

The LTE network is identified by, amongst others, the PLMN-ID. For testing, it is essential to create a new PLMN-ID for the network, so it is clear that the UE is being attached to the correct LTE network and also for legal issues. There are three main operational networks in Portugal, *Vodafone* (26801), *NOS* (26803) and *altice MEO* (26806). The first three digits represent the MCC, 268 for Portugal, and the last digits represent the MNC. As 08 is not attributed to a specific LTE network, it will be used as the setup's MNC. The new PLMN-ID (26808) has to be configured in the OAI eNB, UE, MME and SPGW nodes (every step is described in Appendix C).

Other parameters need to be set on the EPC side, namely the integrity and ciphering algorithms to be used, the IP network settings and the MySQL server configurations. On the eNB side, parameters like the duplex mode (TDD/FDD), E-UTRA frequency band (specifying the downlink and uplink frequencies), number of LTE resource blocks, transmission and reception gain, reference signal power and IP network settings have to be configured. For transmission in unstandardized and unlicensed bands, or in bands not programmed in OAI, such as FDD's Band 31, additional changes have to be made in the source code of the eNB and UE, otherwise the system will not recognize them, all described in Appendix C.

The UE emulates an USIM card, configured using a single file. This file contains information regarding the known PLMNs to the UE, as well as the typical UE and USIM card parameters, IMSI, IMEI, K and OPc. These values are represented in Table 6.1.

Table 6.1: Emulated USIM card for the UE.

	Configuration File USIM Card
K	8baf473f2f8fd09487cccbd7097c6862
OP	11111111111111111111111111111111
OPc	8e27b6af0e692e750f32667a3b14605d
IMSI	268080000000009
IMEI	356092040793012

The UEs will vary depending on the setup (represented by *(a)* and *(b), in Figure 6.1*), but the eNB and EPC remain the same. They're ran on separate computers and are both connected by an Ethernet cable. For the EPC, the following network settings are applied.

- Ethernet port 0, `eth0`: connection to the outside world (Internet), with IPv4 address 192.168.71.105/24, MTU size of 1500.
- Ethernet port 1, `eth1`: connection to the eNB's computer, with IP 192.168.1.16/24 and Maximum Transmission Unit (MTU) size of 1536.

Furthermore, in the SPGW node, a pool of IPv4 addresses to be assigned to the UEs is configured (172.16.0.0/12), with the first address reserved for the GTP network device (172.16.0.1). The network's DNS servers communicated to the UEs are 193.136.92.73 and 193.136.92.74.

On the eNB side, only one port is required to communicate with the EPC, but the setup's computer has two Ethernet ports, so it will be used for a different purpose.

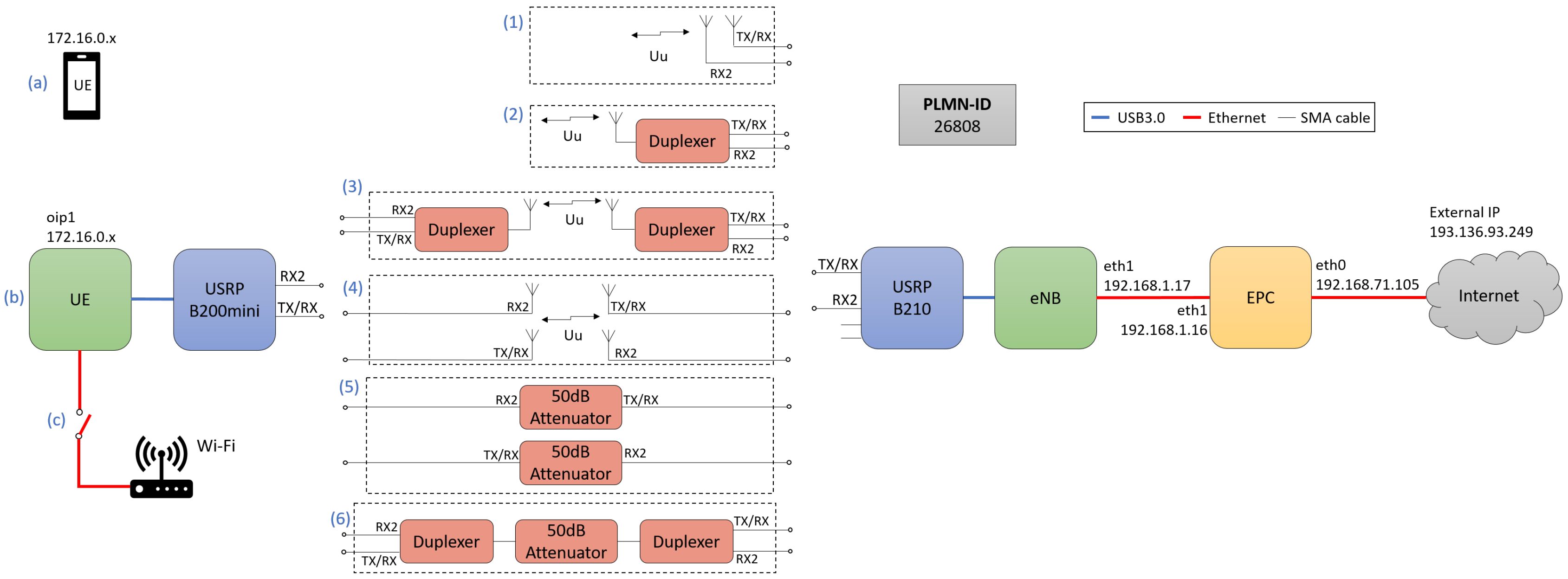


Figure 6.1: Final implementation setup, including two different UEs, various radio configurations, the possibility to share mobile data through a Wi-Fi hotspot, and the IPs assigned to each component.

- Ethernet port 1, `eth1`: connection to the EPC's computer, with IP 192.168.1.17/24 and MTU size of 1536.
- Ethernet port 0, `eth0`: this port is only used for the SBC setup, to establish an SSH remote connection with the board running the OAI UE, saving computing resources by eliminating the graphical interface processing. The configured IP is 192.168.2.17/24 and MTU size of 1536.

The UE won't require any prior IP address configuration, unless it is running in the *UP Squared* SBC. If so, one of its Ethernet ports will be configured with the IPv4 address 192.168.2.16/24 and connected to the eNB, to be remotely controlled.

6.2.1.1 Commercial UEs

Aside from the UEs implemented with software radio, two commercial UEs (smartphones, in this case) are used for testing and are represented below:

- Samsung Galaxy S4 (GT-i9295) [Sam17].
- OnePlus 3 (A3003) [One17].

Table 6.2 depicts the frequency bands recognized by the two smartphones, in both TDD and FDD duplex modes. Some of these will be used for testing, as is later discussed in 6.2.3. In order to connect the UEs to the network, one can't use a regular network operator's USIM card, as some parameters (K, OP) are only known by the network operator itself and can't be added into OAI's subscriber database by a regular user. Therefore, a programmable USIM card has to be used, so every parameter will be programmed by the user.

Table 6.2: Frequency bands supported by the commercial UEs.

	FDD												**TDD**			
Frequency Bands	1	2	3	4	5	7	8	12	17	20	28	30	38	39	40	41
Samsung Galaxy S4	✓		✓		✓	✓	✓			✓						
OnePlus 3	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓

USIM cards The parameters for the USIM cards used in the commercial UEs are in Table 6.3 and were also inserted in the HSS database. The programming of these is described in Appendix C.

6.2.1.2 Photos

In this section, photos of the setup are shown. Figure 6.2 shows the setup for the UE running in a single-board computer. On the UE label, the USRP B200mini is also included, as it joined together with the *UP Squared* in a 3D printed holder (development process in Appendix D). Figure 6.3 shows the used RF components, each being labeled in the photo except for the RF attenuators.

[1] Service Provider Name (SPN)

Table 6.3: USIM cards parameters: card 1 is used on *Samsung Galaxy S4* and card 2 on the *OnePlus 3*.

	Card 1	**Card 2**
K	8baf473f2f8fd09487cccbd7097c6862	00112233445566778899aabbccddeeff
OP	11111111111111111111111111111111	01020304050607080910111213141516
OPc	8e27b6af0e692e750f32667a3b14605d	0ed47545168eafe2c39c075829a7b61f
IMSI	268080000000002	26808000000003
SPN[1]	OpenAirInterface	OpenAirInterface

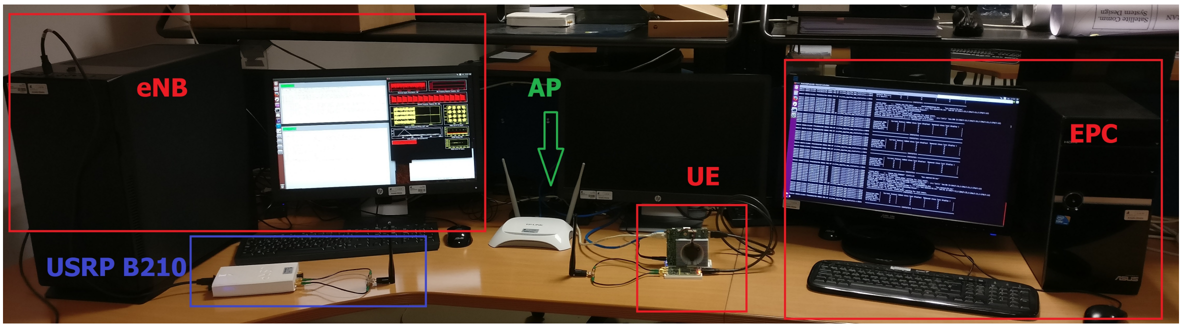


Figure 6.2: Setup for the UE running in a single-board computer. On the UE label, the USRP B200mini is also included.

6.2.2 Tools

In this section, the tools used for obtaining the results are described.

6.2.2.1 Throughput and Latency

The throughput is the most important result, as it gives an estimate of how well the setup is working and can also be used for debugging: for example, if the mobile UDP throughput is so low that it does not allow Internet browsing, the system's radio parameters have to be reconfigured. There are several ways to measure the throughput for UDP and IP. The ones used in this dissertation are `iperf3`, which is executed in the terminal, [ipe17] and *Speedtest.net* by Ookla [Spe17], which makes use of a web browser.

To measure UDP throughputs with `iperf3`, one has to run an `iperf3 server` on EPC's computer, and then run an `iperf3 client` on the UE side (using an `iperf` app for the commercial UEs and Linux's terminal for the rest). This will return the uplink throughput. To obtain the downlink throughput, a reverse flag (`‘-R’`) must be appended to the command, causing the EPC and UE to switch roles. For every measurement, the transmitted packets' bandwidth was started with an initial value of 2 MHz, by appending a `‘-b2M’` flag to the command line, and then incrementing it by 2 MHz at a time until the maximum throughput value was attained.

With *Speedtest.net*, the website (or mobile app, when using the commercial UEs) is accessed and the throughput test is ran by just pressing a button in the center of the screen, which then outputs the downlink/uplink throughputs, along with a latency measurement. This method is not used for the implementation of the UE on a SBC, as no graphical interface is available.

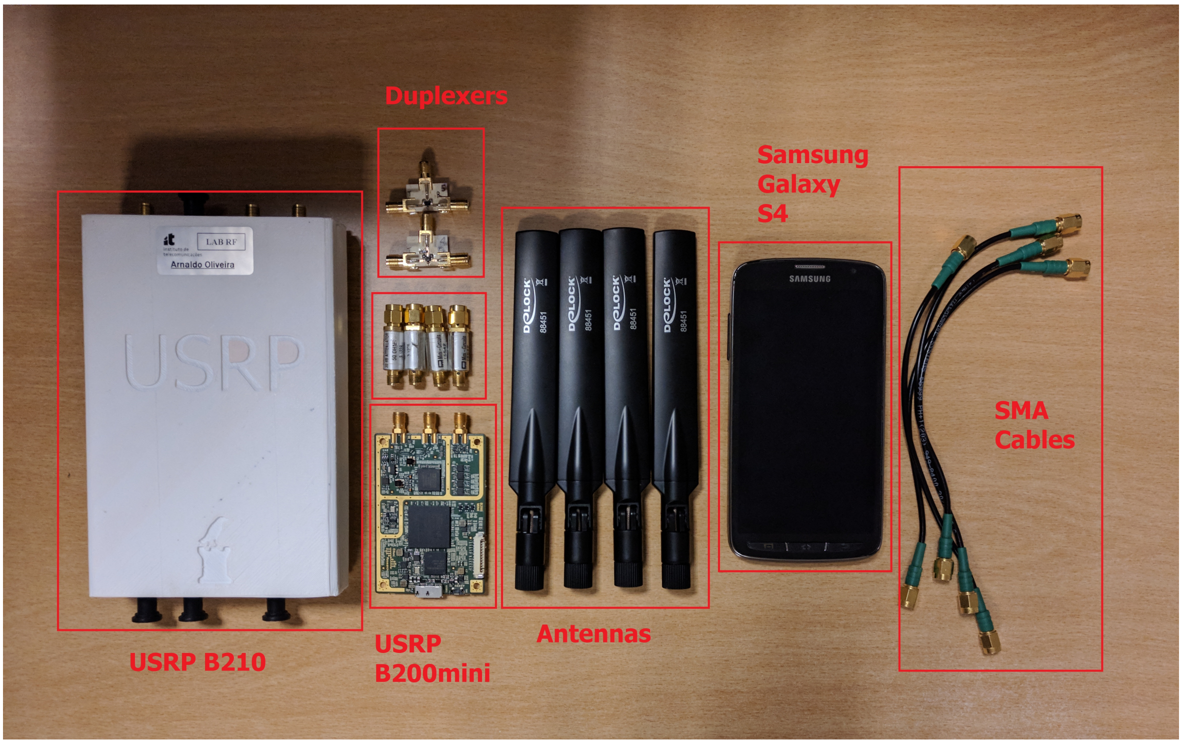


Figure 6.3: Some of the RF components used on the setup. The ones not labeled are the RF attenuators.

Regarding latency, the `ping` command is used to measure the Round-Trip Time (RTT), in milliseconds. The geographic location that was pinged was Castelo Branco, Portugal, distanced approximately 200 km away from the testing location, which was Aveiro, Portugal. It was selected for consistency, due to it being the most picked location by the automatic *Speedtest.net*.

6.2.2.2 Software oscilloscope (Softscope)

OpenAirInterface comes with a physical layer monitoring software oscilloscope (usually referred to as softscope) for both eNB and UE, which plots the received signal power in the time-domain (dB), channel impulse response, channel frequency response, as well as log-likelihood ratios, I/Q components (e.g. QPSK constellations) and throughputs. On the eNB, the log-likelihood ratios, I/Q components and throughput plots are related to the Physical Uplink Shared Channel (PUSCH), while on the UE side the log-likelihood ratios and I/Q components plots are related to the Physical Broadcast Channel (PBCH), Physical Downlink Control Channel (PDCCH) and Physical Downlink Shared Channel (PDSCH), with the last physical channel also having a throughput plot.

6.2.2.3 Logging and packet analysis

A low-level logging tool is provided by OAI, being useful for debugging, code testing and checking the protocols' functionalities of the various network nodes (eNB, UE, SPGW, HSS and MME). The logger allows the user to visualize and store every message exchanged during the setup's execution, according to the verbosity level set in the configuration files.

Along with the logging functionality, Wireshark [Wir17] is also used. It's a free and open-source network protocol analyser, recording every exchanged packet in a selection of network interfaces. Integrated filtering and sorting options are offered, making it useful to examine a group of exchanged messages related to a specific protocol, e.g. S1AP.

6.2.2.4 Power Headroom Report (PHR) and Channel Quality Indicator (CQI)

Power Headroom Report (PHR) belongs to the MAC layer and is used for uplink power control. It indicates how much transmission power is left for a UE to use, in addition to the power being used by the current transmission. If the value is positive, it means there is still room for more data transmission (transmit power can be increased). If it is negative, it means the UE is transmitting more than it should.

The Channel Quality Indicator (CQI) carries information on how good or bad the channel is. In LTE, there are 15 different CQI values, going from 1 to 15. The bigger the value, the better the channel quality. A CQI of 1-6 allows for the use QPSK modulation, 7-9 for 16-QAM and 10-15 for 64-QAM modulation.

The PHR and CQI are reported periodically in the eNB for as long as the UE is attached to the network. However, this is not implemented according to the 3GPP standard in OAI yet, so a scheduler that periodically grants some uplink resources is used for reporting the two, instead. This grant is also used to detect and remove "dead" UEs that were attached to the network but stopped responding after a certain period.

As their report is periodical, their values will change along the period the UE is attached to the network. For example, when the UE is transmitting information (uplink), the reported PHR value will be smaller than when no transmission is happening. The ideal and maximum values for both variables in idle mode are 40 dB for the PHR and 15 for the CQI.

6.2.3 LTE frequency bands

From the wide array of LTE frequency bands available, a group of frequencies was selected for testing. This selection is composed of frequencies that work with the FDD duplexing mode - bands 1, 5, 7 and 31 - and with the TDD duplexing mode - bands 38 and 40. The frequency ranges of these bands were presented in section 2.6.

The FDD bands 1, 5 and 7 are widely used by mobile network operators and were chosen for that exact reason, to test if the UE could run with the most common network configurations. As discussed earlier, in section 5.2.3, these are also the same bands for whom the passive SAW duplexers were designed to.

FDD band 31 is not commonly used (it wasn't even configured in OAI's source code) and is the lowest LTE frequency band on the spectrum. This means it covers a wider area than the usual frequencies, which makes it ideal to implement on remote zones. It's only suitable for 5 MHz bandwidth operation and works on the same frequency range as one of the TETRA bands in Europe, used for emergency services. Commercial UEs don't recognize it, so it can only be tested with the software-defined radio UEs.

TDD deployments are not as common as FDD. TDD band 38 is implemented in Portugal by one carrier, Vodafone, but it is not widely used. TDD band 40 was chosen for testing because in Portugal it is not used for LTE-related purposes, so it would be interesting to take advantage of that premise.

Every subsequent test is performed using 25 resource blocks, i.e. 5 MHz bandwidth. It would be impossible to transmit in band 31 if more resource blocks were used.

6.3 Baseline with Commercial UEs

With OAI's platform, the most stable and reliable implementation of an LTE small-cell scenario is when commercial UEs (smartphones, usually) are used to connect to the network. For testing, two different commercial UEs were used, the *Samsung Galaxy S4* and the *OnePlus 3*, as was described earlier in section 6.2.1.1. The first supports FDD frequency bands only, while the latter also recognizes TDD bands. As this is the most optimized implementation, it will be used as a baseline for the throughput and performance achieved by the other UE deployments.

For the commercial UEs, all bands but band 31 were tested, one at a time. Distinct eNB radio configurations were used for the different frequency bands, described here:

- FDD bands 1, 5 and 7: duplexer and antenna. To each of the developed duplexers was added an antenna, connected to its antenna port, and its transmit/receive ports were connected to the eNB's *USRP B210* TX/RX and RX2 ports, respectively.
- FDD band 31 and TDD bands 38 and 40: two antennas. To each eNB's *USRP B210* TX/RX and RX2 ports was connected one antenna. As no duplexer for band 31 was developed, two antennas were used instead. TDD should be working with one antenna only (connected to the native TX/RX port) and make use of the board's RF switch, but this is not yet implemented in OAI, so one antenna on each port has to be used instead. Ideally, a circulator should be utilized to increase the quality of the transmission and avoid the noise caused by other frequencies, but none was available.

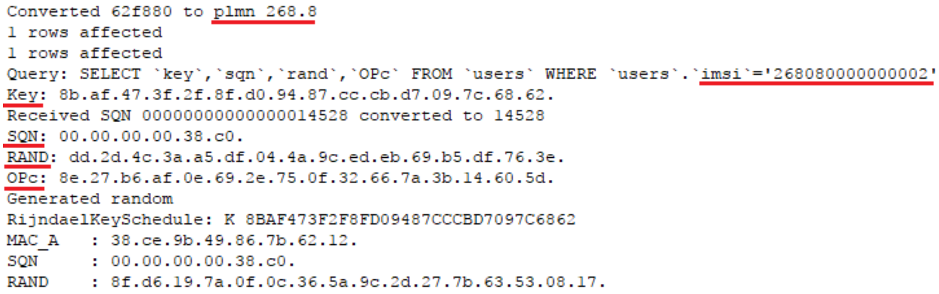
The commercial UEs were tested at a distance of around 1 meter from the setup's base station.

In the following sections, the test results for the commercial UEs will be presented. First, proof of authentication/attach to the network is provided (in section 6.3.1), followed by the throughput, latency and PHR/CQI test results (in section 6.3.2), ending with the software oscilloscopes that allow for monitoring of the network during its execution (in section 6.3.3).

6.3.1 Proof of Authentication/Attach

The UE attach process is independent of the different frequency bands or radio configurations. In order to be attached to the network, the UE must first go through an authentication process, described in more detail in Appendix A. After every step is completed, the UE is successfully attached.

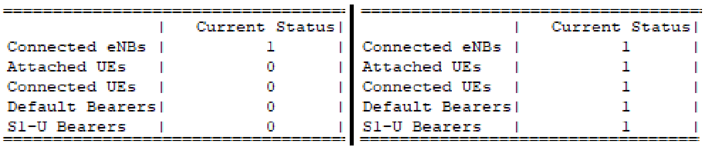
When trying to attach, the UE will communicate its IMSI to the HSS, where the keys and USIM card parameters used for comparison are stored. By receiving the IMSI from the UE, as shown in the excerpt of HSS's log in Figure 6.4, the HSS will use it to search in its database for the previously stored parameters (K, SQN, RAND and OPc), generate/compute new values and then compare them with the UE's. If they are equal, the authentication is passed. The USIM card parameters in the figure are the same as the ones programmed on the USIM card 1, displayed in Table 6.3, which is used in the *Samsung Galaxy S4*.



```
Converted 62f880 to plmn 268.8
1 rows affected
1 rows affected
Query: SELECT `key`,`sqn`,`rand`,`OPc` FROM `users` WHERE `users`.`imsi`='268080000000002'
Key: 8b.af.47.3f.2f.8f.d0.94.87.cc.cb.d7.09.7c.68.62.
Received SQN 00000000000000014528 converted to 14528
SQN: 00.00.00.00.38.c0.
RAND: dd.2d.4c.3a.a5.df.04.4a.9c.ed.eb.69.b5.df.76.3e.
OPc: 8e.27.b6.af.0e.69.2e.75.0f.32.66.7a.3b.14.60.5d.
Generated random
RijndaelKeySchedule: K 8BAF473F2F8FD09487CCCBD7097C6862
MAC_A    : 38.ce.9b.49.86.7b.62.12.
SQN      : 00.00.00.00.38.c0.
RAND     : 8f.d6.19.7a.0f.0c.36.5a.9c.2d.27.7b.63.53.08.17.
```

Figure 6.4: Excerpt of HSS's log, for the commercial UE's authentication/attach process.

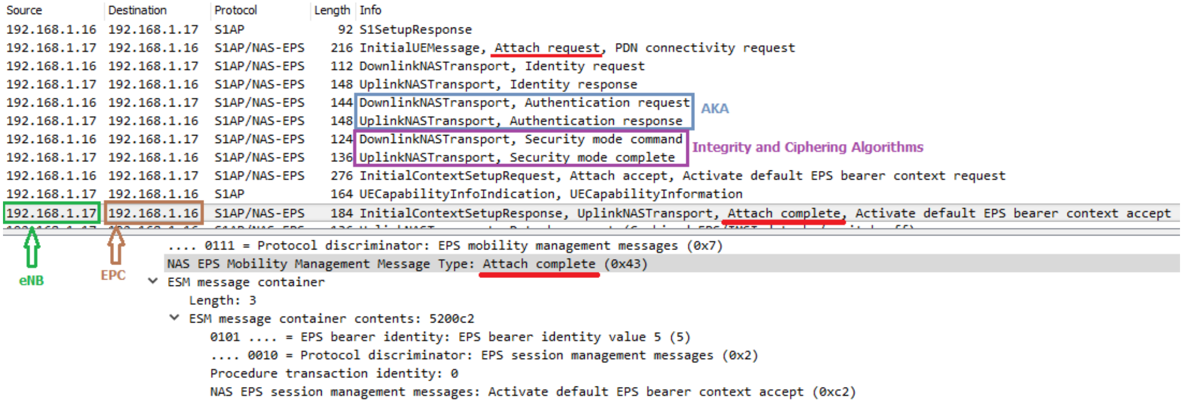
During the attach procedure, the MME will attribute default and S1-U bearers to the UE. After it is successfully attached and connected to the network, the MME will confirm it by reporting the table on the right side of Figure 6.5 on its logs. The left side of Figure 6.5 represents the LTE network waiting for an attach. This can be concluded because, as shown, one eNB is connected to the core network.



```
==================================== ====================================
                |  Current Status|                 |  Current Status|
Connected eNBs  |       1        | Connected eNBs  |       1        |
Attached UEs    |       0        | Attached UEs    |       1        |
Connected UEs   |       0        | Connected UEs   |       1        |
Default Bearers|       0        | Default Bearers|       1        |
S1-U Bearers    |       0        | S1-U Bearers    |       1        |
==================================== ====================================
```

Figure 6.5: Excerpt of MME's logs, for the commercial UE's authentication/attach process. On the left, before the attach procedure. On the right, after the attach procedure completion.

Using Wireshark, the S1AP packets involved in the network attach process can be recorded and filtered. As shown in Figure 6.6, it starts with an Attach request message coming from the eNB to the EPC, going through the Evolved Packet System (EPS) Authentication Key and Agreement (AKA) and security procedures, and finalizing with the Attach complete message.



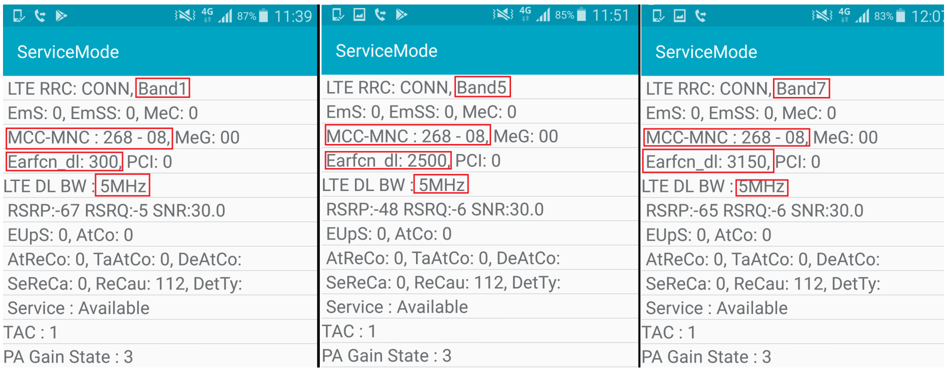
```
Source       Destination  Protocol     Length Info
192.168.1.16 192.168.1.17 S1AP            92 S1SetupResponse
192.168.1.17 192.168.1.16 S1AP/NAS-EPS   216 InitialUEMessage, Attach request, PDN connectivity request
192.168.1.16 192.168.1.17 S1AP/NAS-EPS   112 DownlinkNASTransport, Identity request
192.168.1.17 192.168.1.16 S1AP/NAS-EPS   148 UplinkNASTransport, Identity response
192.168.1.16 192.168.1.17 S1AP/NAS-EPS   144 DownlinkNASTransport, Authentication request
192.168.1.17 192.168.1.16 S1AP/NAS-EPS   148 UplinkNASTransport, Authentication response
192.168.1.16 192.168.1.17 S1AP/NAS-EPS   124 DownlinkNASTransport, Security mode command
192.168.1.17 192.168.1.16 S1AP/NAS-EPS   136 UplinkNASTransport, Security mode complete
192.168.1.16 192.168.1.17 S1AP/NAS-EPS   276 InitialContextSetupRequest, Attach accept, Activate default EPS bearer context request
192.168.1.17 192.168.1.16 S1AP           164 UECapabilityInfoIndication, UECapabilityInformation
192.168.1.17 192.168.1.16 S1AP/NAS-EPS   184 InitialContextSetupResponse, UplinkNASTransport, Attach complete, Activate default EPS bearer context accept

    .... 0111 = Protocol discriminator: EPS mobility management messages (0x7)
    NAS EPS Mobility Management Message Type: Attach complete (0x43)
  ˅ ESM message container
      Length: 3
    ˅ ESM message container contents: 5200c2
        0101 .... = EPS bearer identity: EPS bearer identity value 5 (5)
        .... 0010 = Protocol discriminator: EPS session management messages (0x2)
        Procedure transaction identity: 0
        NAS EPS session management messages: Activate default EPS bearer context accept (0xc2)
```

Figure 6.6: Wireshark S1AP authentication/attach procedure packets of the commercial UE, *Samsung Galaxy S4*.

6.3.1.1 Frequency band confirmation

This section aims to prove the flexibility of the network implementation regarding the transmission on different frequency bands, by showing the commercial UEs have attached to the tested frequency bands.

The *Samsung Galaxy S4* has a built-in Service Mode that contains relevant informations about the network, one of them being the frequency band it is connected to, as well as the MCC/MNC and the used bandwidth (LTE DL BW - number of resource blocks). It is accessed by executing the USSD[1]code $\star$#0011# on the phone. Figure 6.7 represents three screenshots of the UE's Service Mode for bands 1, 5 and 7. The Mobile Country Code (MCC) is 268, the Mobile Network Code (MNC) is 08 and the used bandwidth is 5 MHz (*LTE DL BW*), confirming the use of 25 LTE resource blocks.



11:39

ServiceMode

LTE RRC: CONN,Band1
EmS: 0, EmSS: 0, MeC: 0
MCC-MNC : 268 - 08, MeG: 00
Earfcn_dl: 300, PCI: 0
LTE DL BW : 5MHz
RSRP:-67 RSRQ:-5 SNR:30.0
EUpS: 0, AtCo: 0
AtReCo: 0, TaAtCo: 0, DeAtCo:
SeReCa: 0, ReCau: 112, DetTy:
Service : Available
TAC : 1
PA Gain State : 3

11:51

ServiceMode

LTE RRC: CONN,Band5
EmS: 0, EmSS: 0, MeC: 0
MCC-MNC : 268 - 08, MeG: 00
Earfcn_dl: 2500, PCI: 0
LTE DL BW : 5MHz
RSRP:-48 RSRQ:-6 SNR:30.0
EUpS: 0, AtCo: 0
AtReCo: 0, TaAtCo: 0, DeAtCo:
SeReCa: 0, ReCau: 112, DetTy:
Service : Available
TAC : 1
PA Gain State : 3

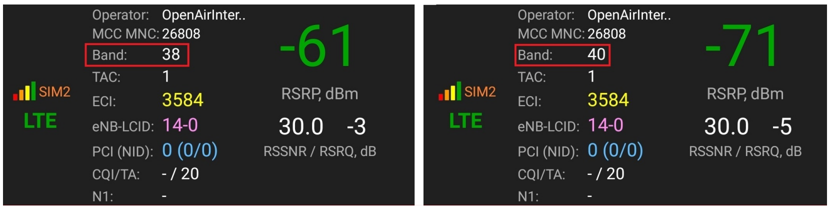
12:0

ServiceMode

LTE RRC: CONN,Band7
EmS: 0, EmSS: 0, MeC: 0
MCC-MNC : 268 - 08, MeG: 00
Earfcn_dl: 3150, PCI: 0
LTE DL BW : 5MHz
RSRP:-65 RSRQ:-6 SNR:30.0
EUpS: 0, AtCo: 0
AtReCo: 0, TaAtCo: 0, DeAtCo:
SeReCa: 0, ReCau: 112, DetTy:
Service : Available
TAC : 1
PA Gain State : 3

Figure 6.7: *Samsung Galaxy S4*'s Service Mode. From left to right: bands 1, 5 and 7.

TDD is only supported by the *OnePlus 3*. It doesn't have a secret service mode like the *Galaxy S4*'s, but several apps can be used to discover which frequency band the phone is connected to. In this case, the *Network Cell Info Lite* app [Net17] was used and it is possible to confirm the phone correctly attaches to bands 38 and 40 in Figure 6.8. The operator's name that was given to the network is also shown, *OpenAirInterface*, along with the global network identifiers MCC (268) and MNC (08).



SIM2
LTE
Operator: OpenAirInter..
MCC MNC: 26808
Band: 38
TAC: 1
ECI: 3584
eNB-LCID: 14-0
PCI (NID): 0 (0/0)
CQI/TA: - / 20
N1: -
-61
RSRP, dBm
30.0 -3
RSSNR / RSRQ, dB

SIM2
LTE
Operator: OpenAirInter..
MCC MNC: 26808
Band: 40
TAC: 1
ECI: 3584
eNB-LCID: 14-0
PCI (NID): 0 (0/0)
CQI/TA: - / 20
N1: -
-71
RSRP, dBm
30.0 -5
RSSNR / RSRQ, dB

Figure 6.8: *OnePlus 3* connection to bands 38 and 40, using the *Network Cell Info Lite* app.

6.3.2 Throughput, latency and PHR/CQI results

The UDP throughput and latency test results, together with the PHR and CQI values, are presented in this section and represented in Table 6.4. Both phones were tested. The

[1] Unstructured Supplementary Service Data

results for the *Samsung Galaxy S4* were obtained using the *Speedtest.net* mobile app, while for the *OnePlus 3* they were obtained using an *iperf3* app. The two different test methods were used because when trying to obtain results on the *OnePlus* using the *Speedtest.net* app, the eNB started crashing and the phone dropped the connection to the network. This is a common problem when testing with newer phones and is caused by the UE's necessity of a better timing reference than the one obtained by just using the USRP. To solve this, a GPS[2]reference signal module can be added to the SDR hardware.

Table 6.4: Commercial UEs uplink and downlink throughput, latency, PHR and CQI measured results.

	UEs	**DL(Mbps)**	**UL(Mbps)**	**Latency(ms)**	**PHR(dB)**	**CQI**
Band 1	Galaxy S4	15.29	9.05	27.0	40	15
	OnePlus 3	15.6	7.65	24.0	40	15
Band 5	Galaxy S4	14.69	8.56	27.0	40	15
	OnePlus 3	15.9	7.27	26.3	40	15
Band 7	Galaxy S4	14.22	8.54	23.0	40	15
	OnePlus 3	16.3	7.93	25.0	40	15
Band 38	OnePlus 3	3.93	2.23	29.8	40	13
Band 40	OnePlus 3	4.39	1.95	29.4	40	14

The throughput is very similar for all FDD frequency bands and smartphones, with FDD oscillating around 16/7.5 Mbps on the *OnePlus* and 14.7/8.6 Mbps on the *Galaxy S4*. Between the TDD frequency bands, the results are also similar, oscillating around 4.2/2 Mbps. The latency is very similar for every band, but with TDD having a worse performance.

There is a clear discrepancy between the FDD results and the TDD results. Starting with the PHR and CQI samples, in FDD they are always at their maximum/ideal level, PHR 40 dB and CQI 15. In TDD, however, the Channel Quality Indicator (CQI) is lower. Another visible issue is TDD's throughput. When compared to FDD, the difference is drastic, with it achieving about 25% of FDD's throughput.

These results can be explained by the radio differences between the two implementations and the lack of optimization on OAI's TDD source code. As a duplexer with a single antenna was used for the FDD bands, the selectivity is improved considerably and the noise caused by different frequencies is reduced. For the TDD bands' case, on the other hand, two antennas were used without a duplexing agent, meaning there is a lot more noise in the air that is not being attenuated.

TDD is a more recent implementation than FDD, in OAI, and FDD is the main test subject for most of the OAI users. Therefore, there is a bigger interest for the developers to optimize FDD's source code, so TDD ends up being more neglected.

Even though the UDP throughput is lower for TDD, it is still able to browse the Internet, play Youtube videos (with lower quality than in FDD) and make *Skype* calls.

6.3.3 Software oscilloscopes

In this section, the software oscilloscopes (softscopes) that allow for monitoring of the physical layer during the setup's execution are presented. For the commercial UEs, it is

[2] Global Positioning System

only possible to observe these softscopes in the eNB. Depending on the type of transmission (FDD/TDD) and the state it is on (i.e. initial attach state, resource-intensive transmission state and regular transmission state), the monitored values will vary. The **initial attach state** represents the moment the UE attaches to the network. The **resource-intensive transmission state** represents the moments the UE is using its data the most, for example, watching a video on YouTube, making a Skype call or running a throughput test. The **regular transmission state** represents the uplink transmissions for the regular use of the UE, such as browsing the Internet or for network signalling. As the softscopes were similar between each frequency band and duplex mode used, only one per state of transmission is displayed.

In Figure 6.9, the softscope for regular transmission state is presented, and the most relevant plots are discussed.

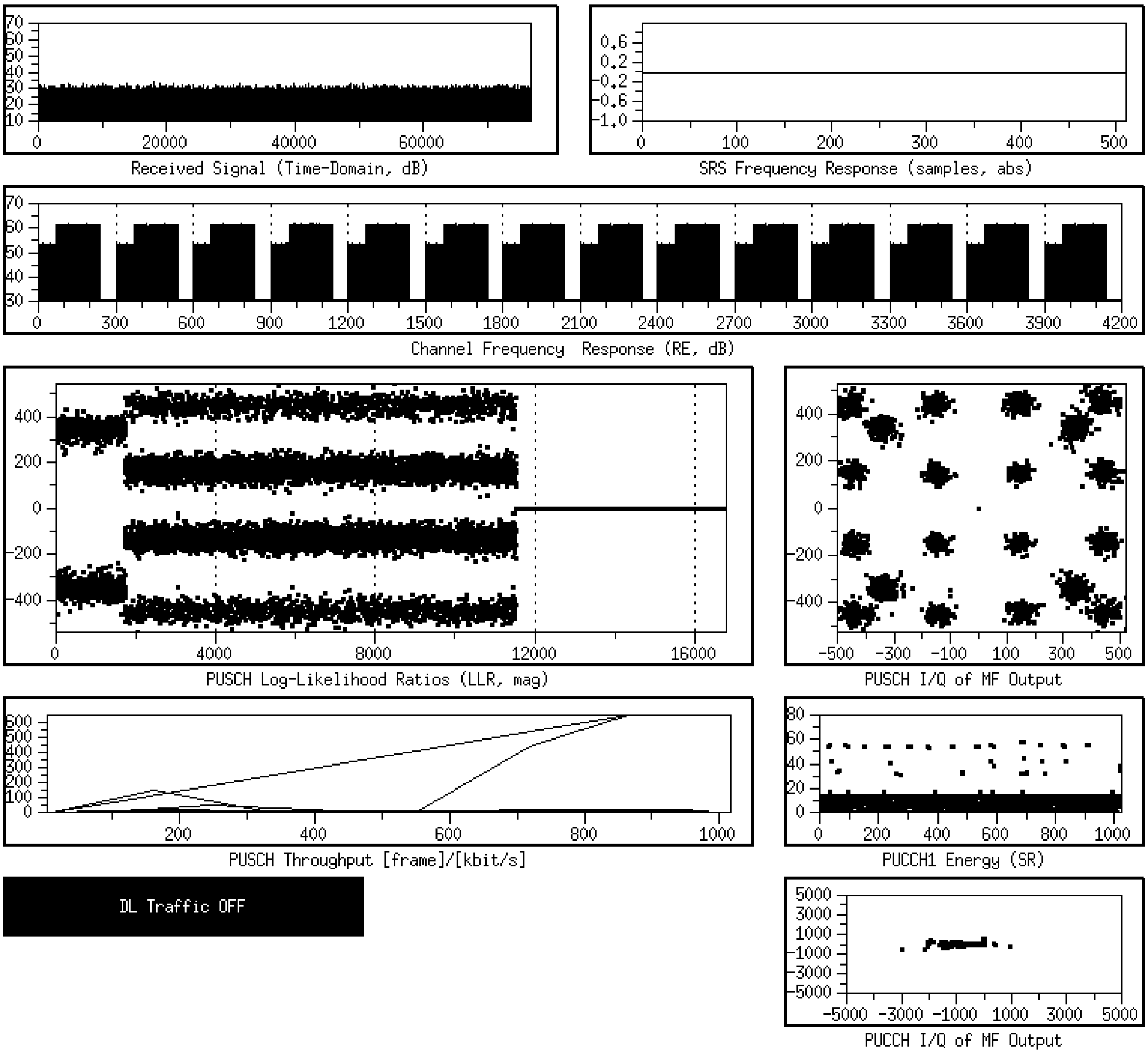


Figure 6.9: Commercial UEs: eNB softscope for the regular transmission state.

Starting from the top, on the left is the received signal in the time-domain. It is flat around the 30 dB mark when no data is being transmitted and occasionally, when signalling or minor data transmission occurs, a spike in the received signal is plotted (not represented in the Figure). Further down is the channel frequency response. Below the channel frequency response are the Physical Uplink Shared Channel (PUSCH) log-likelihood ratios (on the left) and PUSCH I/Q components plot (on the right). The PUSCH constellation for the regular transmission state represents the uplink transmission data modulated with a 16-QAM modulation and a Zadoff-Chu sequence, which comes from the physical signals, usually the PUSCH DeModulation Reference Signal (DMRS). Below these plots are the PUSCH throughput plot on the left, followed by the Physical Uplink Control Channel (PUCCH) energy on the right and a plot of the PUCCH I/Q components below.

In Figure 6.10, two softscopes for the resource-intensive transmission state are displayed: one for FDD, on the left, and one for TDD, on the right. These softscopes were saved when a throughput test was being executed. The two duplex modes are distinguished by looking at the Received Signal in the time-domain plot (highlighted in red). On the left, for FDD band 7, the received signal is capped at almost 60 dB, which is a clear sign that the eNB is receiving an huge amount of data from the UE (as shown in the previous Figure, when the eNB is not receiving UE data it remains capped around 30 dB). On the right, for TDD band 38, it is clear that the received signal is unstable and its quality is worst than FDD's. The first part is saturating (i.e. the received signal is way too high, over 70 dB), then for a portion of time the eNB is receiving the uplink signal correctly (close to 50 dB), but afterwards it keeps on saturating and oscillating between good and saturated values. This is a clear representation that the radio setup used in this dissertation provides much better results when using a duplexing agent than without. The source code is also less optimized for TDD.

Regarding the PUSCH constellations (also highlighted in red), they are a clear 16-QAM modulation, which is used on the uplink data transmission. The constellation on the right, belonging to TDD, is quite noisy when compared to FDD's.

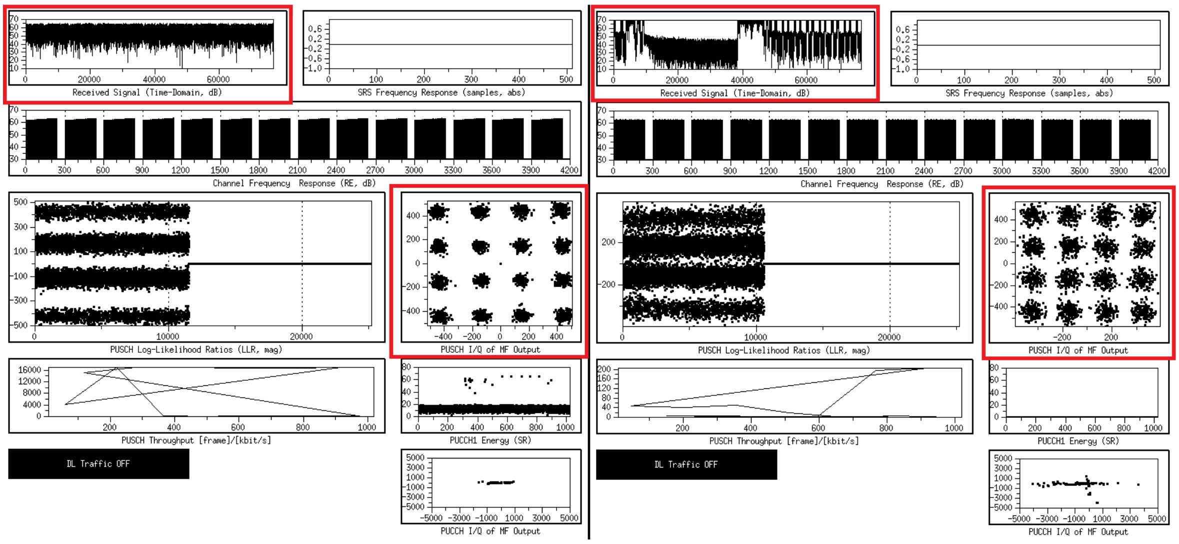

Received Signal (Time-Domain, dB)
SRS Frequency Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
DL Traffic OFF
PUCCH I/Q of MF Output
Received Signal (Time-Domain, dB)
SRS Frequency Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
DL Traffic OFF
PUCCH I/Q of MF Output

Figure 6.10: Commercial UEs: eNB softscopes for the uplink transmission. On the left, FDD band 7. On the right, TDD band 38.

This concludes the discussion regarding the obtained results from testing with the commercial UE implementation.

6.4 UE running on a mini-computer

Now that the reference values were obtained, at least for FDD, a software-defined radio user equipment will be implemented on a mini-computer (*Intel NUC*), and the commercial UE won't be used anymore.

Two kinds of radio implementations were tested: a wired connection between the UE and the eNB and a wireless connection. Different frequency bands and radio configurations were used for the two kinds of implementations, described below:

- Wired connection:
  - FDD band 1: two duplexers and 50 dB RF attenuator. To the developed duplexers was added a 50 dB RF attenuator, connected to their antenna port, and their transmit/receive ports were connected to the *USRPs*' TX/RX and RX2 ports, respectively.
  - FDD band 31 and TDD band 40: two 50 dB RF attenuators. Between each USRP's TX/RX and RX2 ports was connected one 50 dB RF attenuator.
- Wireless connection:
  - FDD bands 1, 5 and 7: two duplexers and two antennas. To each of the developed duplexers was added an antenna, connected to its antenna port, and its transmit/receive ports was connected to the USRPs' TX/RX and RX2 ports, respectively.
  - FDD band 31 and TDD bands 38 and 40: four antennas. To each USRP's TX/RX and RX2 ports was connected one antenna.

The UE was tested at a distance of approximately 1 meter from the setup's base station.

In the following sections, the test results for the UE implemented in a mini-computer will be presented. First, the authentication/attach to the network is discussed (in section 6.4.1), followed by the implementations using a wired (section 6.4.2) and wireless (section 6.4.3) connections between the eNB and the UE, where the throughput, latency and PHR/CQI test results are analysed, ending with the software oscilloscopes that allow for monitoring of the network during its execution (in section 6.4.4).

6.4.1 Authentication/Attach

As was previously mentioned in section 6.3.1, the attach procedure completion does not depend on the frequency band or radio setup; it is equal throughout the different configurations. As the procedure is very similar, it won't be described again. The only difference would be the IMSI used to start the authentication procedure: 268080000000009.

6.4.2 Wired connection between the eNB and the UE

As this setup requires a wired connection between two USRPs, additional care must be taken. The boards should never be directly connected, as that would possibly cause irreversible damage. According to the USRP's users manual, they should be connected with 30 dB of attenuation in between. The amount of noise in the air is around 20 dB of attenuation.

So, to also be able to simulate the air interface in this cabled setup and keep the configured power parameters, 50 dB RF attenuators were used.

The LTE frequency bands used for the subsequent tests for this implementation were chosen so they would represent one type of transmission each: FDD band 1 will represent all the FDD implementations with a duplexer and an attenuator in between; FDD band 31 will represent all FDD implementations with two attenuators between the eNB and UE; TDD band 40 will represent all TDD implementations with two attenuators between the eNB and UE. Because no air interface is used, there's no noise/interference coming from other frequencies in the air, so transmitting in more than one band with the same radio setup would yield redundant results.

First, the regular UDP throughput tests, latency and so on were performed, in section 6.4.2.1. Then, after the network was shared by using a Wi-Fi hotspot, a phone was connected to the free network and performed measurements on the throughput and latency, in section 6.4.2.2.

6.4.2.1 Throughput, latency and PHR/CQI results

The UDP throughput and latency test results, together with the PHR and CQI values, are presented in this section and represented in Table 6.5

Table 6.5: UE running on the *Intel NUC* with a wired connection between the eNB and UE: uplink and downlink throughput, latency, PHR and CQI measured results.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**	**PHR (dB)**	**CQI**
Band 1	7.01	9.05	28.0	22	15
Band 31	2.75	7.85	28.4	34	7
Band 40	9.44	2.36	26.0	32	15

By looking at the LTE UDP throughput, it's clear that band 1 produces the best results of the three, as it is the only band that outputs more than 7 Mbps in both downlink and uplink. This is the setup with duplexers are used, so it is safe to say that even when testing with a cabled setup, a duplexer should be used. Comparing the results obtained with the commercial UEs for band 1 (15.4/8.65 Mbps), the downlink value is lower by more than half of the reference value.

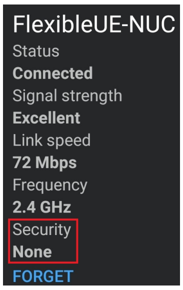
TDD band 40 increased its throughput when compared to the commercial UE's implementation. This proves that the noise caused by the surrounding frequencies when transmitting in the air interface was causing some throughput problems, and this will now be considered the reference value for the TDD throughput results - 9.44/2.36 Mbps.

FDD band 31 was never tested before, so there are no results to compare it with. In terms of the uplink throughput it is working well, but the downlink is too low. The PHR and CQI levels are okay for Band 1 and 40, while Band 31 has a CQI of 7, less than half the ideal value. The latency is similar amongst all.

6.4.2.2 Shared Wi-Fi throughput results

After the UE is attached to the network, the Wi-Fi module in the mini-computer (*Intel NUC*) is used to create a free Wi-Fi hotspot that uses the received LTE signal as input, named *FlexibleUE-NUC*. Proof on the existence of the shared Wi-Fi network can be seen in

Figure 6.11. A device is then connected to it to measure the UDP throughput, displayed in Table 6.6.



FlexibleUE-NUC

Status

Connected

Signal strength

Excellent

Link speed

72 Mbps

Frequency

2.4 GHz

Security

None

Figure 6.11: Free Wi-Fi hotspot created from the received LTE signal, on the *Intel NUC* mini-computer, accessed with the *OnePlus 3* UE.

Table 6.6: UE running on the *Intel NUC* with a wired connection between the eNB and UE: throughput and latency measured via the shared Wi-Fi network.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**
Band 1	9.24	2.36	28.0
Band 31	2.39	8.32	27.0
Band 40	8.33	2.35	27.0

The UDP throughput when using the Wi-Fi hotspot is expected to be around the same value of throughput obtained from the LTE mobile data. This is the case for bands 31 and 40, but not for band 1, where the UL dropped from 9.05 Mbps to 2.36 Mbps.

6.4.3 Wireless connection between the eNB to the UE

In this section, the eNB and UE are connected through the air interface. All of the selected bands were tested.

First, the regular UDP throughput tests, latency and so on were performed, in section 6.4.3.1. Then, after the network was shared using a Wi-Fi hotspot, a phone was connected to the free network and performed measurements on the throughput and latency, in section 6.4.3.2.

6.4.3.1 Throughput, latency and PHR/CQI results

The UDP throughput and latency test results, together with the PHR and CQI values, are presented in this section and represented in Table 6.7.

Table 6.7: UE running on the *Intel NUC* with a wireless connection between the eNB and UE: uplink and downlink throughput, latency, PHR and CQI measured results.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**	**PHR (dB)**	**CQI**
Band 1	7.21	8.06	20.4	39	15
Band 5	4.46	7.81	28.2	37	15
Band 7	6.56	7.27	21.2	32	15

The throughput achieved with LTE data is okay for band 1 and 7, and is a bit low on band 5's downlink. When compared to the reference values obtained with the commercial UEs, the uplink is very similar, but the downlink values decrease by more than half.

As it is possible to notice, only the FDD bands 1, 5 and 7 are depicted in the measured throughputs table. It was not possible to measure the throughputs for FDD band 31 and TDD bands 38 and 40. Although all of them managed to attach to the network, they would get automatically disconnected afterwards. This is, again, an optimization and a radio problem. For these bands, two antennas are used, which worsens the quality of the connection, instead of the duplexer and antenna approach used for bands 1, 5 and 7.

6.4.3.2 Shared Wi-Fi throughput results

After the UE is connected to the created free Wi-Fi hotspot, the throughput is measured and displayed in Table 6.8.

Table 6.8: UE running on the *Intel NUC* with a wireless connection between the eNB and UE: throughput and latency measured via the shared Wi-Fi network.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**
Band 1	7.18	2.19	25.0
Band 5	4.26	5.00	26.0
Band 7	6.62	8.32	29.0

Regarding the Wi-Fi hotspot iperfs, band 5 and band 7 remained close to the original LTE throughput values, while band 1's uplink dropped from 8.06 Mbps to 2.19 Mbps. This also happened in the cabled setup for the same frequency band.

So far, all results stayed behind the commercial UEs data rates. This was expected, considering the software radio UE is not as optimized as a smartphone.

6.4.4 Software oscilloscopes

In this section, the software oscilloscopes (softscopes) that allow for monitoring of the physical layer during the setup's execution are presented. For the software-defined radio UEs' implementation, it is possible to observe the softscopes in the eNB and UE. Depending on the type of transmission (FDD/TDD) and the state they are on, the monitored values will vary. As the softscopes were similar between each frequency band and duplex mode used, only one per state of transmission is displayed.

In Figure 6.12, the softscopes for the initial attach, regular transmission and resource-intensive transmission states are presented, respectively, and their most relevant plots are discussed. The **initial attach** state, on the left, represents the UE right after attaching to the network, when data was yet to be transmitted. Its received signal (highlighted in red) should be and is similar to the one presented in the middle softscope, which represents the **regular transmission** state. As discussed in the section 6.3.3, an occasional spike in the received signal will be plotted, representing the signalling or minor data transmission. The initial attach state PUSCH I/Q components' plot represent a QPSK modulation. The other two softscopes that represent the regular transmission and **resource-intensive transmission** are very similar to the ones obtained in the commercial UE implementation. This is an important aspect, considering the softscope on the right was recorded with TDD band 40 in the cabled setup. When comparing the received signal with the one on Figure 6.10, where the signal is transmitted over the air interface, the cabled setup yields much better results: there is no saturation in the received uplink signal and it remains flat for the whole transmission.

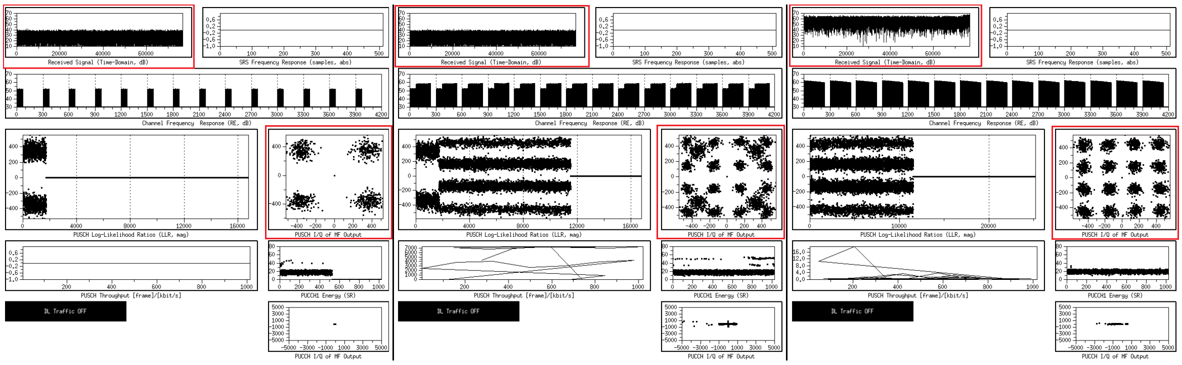

Received Signal (Time-Domain, dB)
SRS Frequency Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
PUCCH I/Q of MF Output
DL Traffic OFF

Figure 6.12: UE running on the *Intel NUC*: eNB softscopes for the initial attach, regular transmission and resource-intensive transmission states, respectively. On the left/middle, FDD band 7. On the right, TDD band 40 (cabled connection).

In Figure 6.13, two UE softscopes representing the downlink signal transmission are displayed: the one on the left represents the transmission for a wireless connection between the eNB and UE (FDD band 1), and the one on the right represents the transmission for a wired connection between the eNB and UE (TDD band 40).

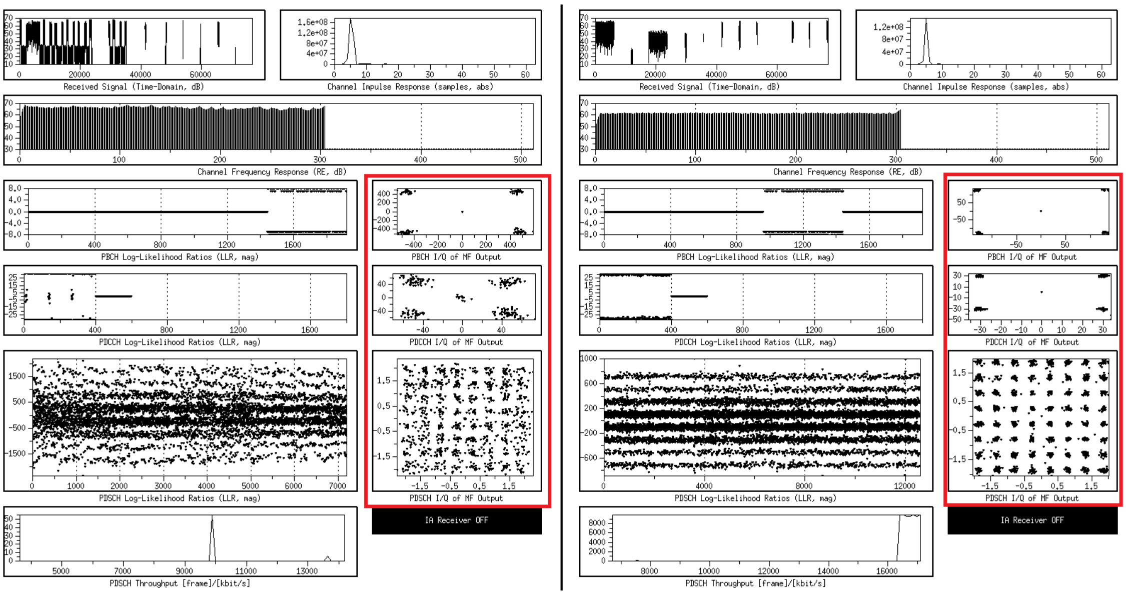

Received Signal (Time-Domain, dB)
Channel Impulse Response (samples, abs)
Channel Frequency Response (RE, dB)
PBCH Log-Likelihood Ratios (LLR, mag)
PBCH I/Q of MF Output
PDCCH Log-Likelihood Ratios (LLR, mag)
PDCCH I/Q of MF Output
PDSCH Log-Likelihood Ratios (LLR, mag)
PDSCH I/Q of MF Output
IA Receiver OFF
PDSCH Throughput [frame]/[kbit/s]

Figure 6.13: UE running on the *Intel NUC*: UE softscopes for the downlink. On the left, FDD band 1. On the right, TDD band 40 (cabled connection).

The UE softscope contains more plots than the eNB's. It starts by plotting the received signal in the time-domain, followed by the channel impulse response. Below, the channel frequency response is plotted. Then, the Physical Broadcast Channel (PBCH), Physical

Downlink Control Channel (PDCCH) and Physical Downlink Shared Channel (PDSCH)'s log-likelihood ratios and I/Q components plots are displayed, one different line for each, finishing with a plot of the PDSCH throughput.

The UE's received signal is not "continuous" like the eNB's. The PBCH, PDCCH and PDSCH constellations for both frequency bands are highlighted in red in Figure 6.13 (from top to bottom). TDD band 40, on the right, provides less noisier constellations than FDD band 1, on the left. PBCH and PDCCH constellations show a QPSK modulation, while the PDSCH represents a 64-QAM used on the downlink signal transmission, during a throughput test.

This completes the discussion of the results obtained for the software-defined radio UE implemented on a powerful mini-computer, the Intel NUC.

6.5 UE running on a Single-Board Computer (SBC)

To transform the UE into a more compact and portable solution, it was decided that it would be migrated from the mini-computer to a single-board computer, the *UP Squared*.

Just like the previous implementations, two kinds of radio implementations were tested: a wired connection between the UE and the eNB and a wireless connection. Different frequency bands and radio configurations were used for the two types of implementations, described below:

- Wired connection:
  - FDD band 1: two duplexers and a 50 dB RF attenuator in between.
  - FDD band 31 and TDD band 40: two 50 dB RF attenuators in between the two USRPs.
- Wireless connection:
  - FDD bands 1, 5 and 7: two duplexers and two antennas, one of each per USRP.
  - FDD band 31 and TDD bands 38 and 40: four antennas, two per USRP.

The UE was tested at a distance of approximately 1 meter from the setup's base station.

In the following sections, the test results for the UE implemented in a single-board computer will be presented. First, the authentication/attach to the network is discussed (in section 6.5.1), followed by the implementations using a wired (section 6.5.2) and wireless (section 6.5.3) connections between the eNB and the UE, where the throughput, latency and PHR/CQI test results are analysed, ending with the software oscilloscopes that allow for monitoring of the network during its execution (in section 6.5.4).

6.5.1 Authentication/Attach

The network configurations haven't changed from the previous UE implementation, so the proof of authentication and complete network attach are in section 6.4.1.

6.5.2 Wired connection between the eNB and the UE

Once again, for a cabled connection between the eNB and the UE's USRPs, 50 dB RF attenuators are used: 30 dB for potential damage protection and 20 dB to simulate the noise in the air interface.

The LTE frequency bands tested in this implementation were the same as in the previous implementation (section 6.4), and the main motivation for their selection was already detailed.

First, the regular UDP throughput and latency testes were performed, in section 6.5.2.1. Then, after the network was shared using a Wi-Fi hotspot, a phone was connected to the free network and performed measurements on the throughput and latency, in section 6.5.2.2.

6.5.2.1 Throughput, latency and PHR/CQI results

The UDP throughput and latency test results, together with the PHR and CQI values, are presented in this section and represented in Table 6.9.

Table 6.9: UE running on the *UP Squared* with a wired connection between the eNB and UE: uplink and downlink throughput, latency, PHR and CQI measured results.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**	**PHR (dB)**	**CQI**
Band 1	3.4	4.72	21.0	20	15
Band 31	2.49	7.01	26.0	35	7
Band 40	9.51	2.10	25.2	34	15

With the same radio conditions, the UE implemented on the *UP Squared* achieves the same throughput and PHR/CQI as the UE implemented on the *NUC* for FDD band 31 and TDD band 40.

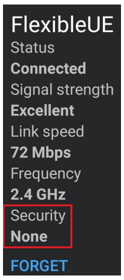
In FDD band 1, however, the throughput is reduced by half. The *UP Squared* has much less computational power than a regular computer, so a greater amount of samples are lost. This means the overall throughput levels are also expected to be lower than for the regular implementation.

6.5.2.2 Shared Wi-Fi throughput results

After the UE is attached to the network, an AP's LAN port is connected to one of the single-board computer's Ethernet port. After configuration, it is used to create a free Wi-Fi hotspot that uses the received LTE signal as input, named *FlexibleUE*. Proof on the existence of the shared Wi-Fi network can be seen in Figure 6.14. A device is then connected to it to measure the UDP throughput, displayed in Table 6.10.

Table 6.10: UE running on the *UP Squared* with a wired connection between the eNB and UE: throughput and latency measured via the shared Wi-Fi network.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**
Band 1	4.19	4.85	20.0
Band 31	2.34	8.11	25.0
Band 40	9.02	1.93	21.0



FlexibleUE
Status
Connected
Signal strength
Excellent
Link speed
72 Mbps
Frequency
2.4 GHz
Security
None

Figure 6.14: Free Wi-Fi hotspot created from the received LTE signal, detected with the *OnePlus 3* UE.

The rates obtained from the Wi-Fi hotspot shared by the AP are very close to the original LTE network rates, so everything is working as expected. Band 1 provides even higher throughput than what was measured before sharing the network.

6.5.3 Wireless connection between the eNB and the UE

In this section, the eNB and UE are connected through the air interface. All of the selected bands were tested.

First, the regular UDP throughput and latency tests were performed, in section 6.5.3.1. Then, after the network was shared by means of a Wi-Fi hotspot, a phone was connected to the free network and performed measurements on the throughput and latency, in section 6.5.3.2.

6.5.3.1 Throughput, latency and PHR/CQI results

The UDP throughput and latency test results, together with the PHR and CQI values, are presented in this section and represented in Table 6.11.

Table 6.11: UE running on the *UP Squared* with a wireless connection between the eNB and UE: uplink and downlink throughput, latency, PHR and CQI measured results.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**	**PHR (dB)**	**CQI**
Band 1	5.64	6.82	17.9	37	15
Band 5	1.18	8.11	29.4	35	13
Band 7	5.96	7.34	26.8	33	15

The throughput results are worse than the ones obtained with the *NUC* mini-computer. This was expected, especially when comparing the processing power of both machines. However, the difference is not astounding, meaning that the migration of the UE into a more compact solution was successful.

Regarding the uplink throughput, the three frequency bands remained close to the reference value obtained with the commercial UEs. The downlink throughput, however, only reached about one third of the total throughput achieved by the smartphones for FDD bands 1 and 7, and much worse for band 5.

For over-the-air transmission, only the FDD bands 1, 5 and 7 work as expected, just like in the *Intel NUC* implementation. FDD band 31 and TDD bands 38 and 40 manage to attach to the network but get automatically disconnected, and their radio performance is very poor.

6.5.3.2 Shared Wi-Fi throughput results

As was said earlier, an Access Point (AP) is used to share the received signal with surrounding users. After a UE is connected to the created free Wi-Fi hotspot, the UDP throughput is measured and displayed in Table 6.12.

Table 6.12: UE running on the *UP Squared* with a wireless connection between the eNB and UE: throughput and latency measured via the shared Wi-Fi network.

	DL (Mbps)	**UL (Mbps)**	**Latency (ms)**
Band 1	8.36	7.21	28.0
Band 5	8.17	0.5	24.0
Band 7	4.98	8.26	29.0

Regarding the Wi-Fi throughput results, FDD band 1 is impressive, considering it surpasses the values obtained in any implementation (except for the commercial UEs).

For FDD band 5, the uplink's 0.5 Mbps might be caused by the band 5's duplexers bad design, specified in Appendix B.

6.5.4 Software oscilloscopes

In this section, the software oscilloscopes (softscopes) used for the monitoring of the PHY layer during the setup's execution are presented. As the single-board computer *UP Squared* was accessed remotely, in order to not waste any computing power, only the eNB softscopes can be visualized. As was referred in the softscope section of the previous UE implementation, the obtained plots were very similar amongst the different frequency bands, so only the relevant ones will be presented.

Figure 6.15 represents the **initial attach** state, after the UE's attach but when no data has been transmitted yet, only signalling messages.

On the left, FDD band 7 is represented, with TDD band 40 (cabled connection) on the right. The red highlighted areas represent the received signal in the time domain and the PUSCH constellations for both softscopes. Band 7's received signal plot (around 35 dB) is more stable than TDD band 40's (cabled connection). The latter is more attenuated than the band 7's, with its received signal plot being flat at around 25 dB (10 dB difference). The spikes caused by the transmission of signalling messages are also visible. Band 40's channel frequency response, highlighted in blue, is also more attenuated than band 7's. Both PUSCH constellations represent a QPSK modulation. Although it appears the QPSK modulation for the band 40 is shifted, it is just that the plot scale is different.

Figure 6.16 represents the regular transmission and resource-intensive transmission states for FDD band 1 (cabled connection) and FDD band 7.

Both softscopes look stable and similar to previously obtained results. On the left side and highlighted in red, the spike in the received signal represents the transmission of signalling or of minor data. Also on the left, the regular transmission state's PUSCH constellation represents the expected 16-QAM modulation from uplink data, shared with a Zadoff-Chu sequence from signalling. On the right side of the Figure, the received uplink signal is flat through the whole transmission (at around 60 dB) and is never saturated. The PUSCH

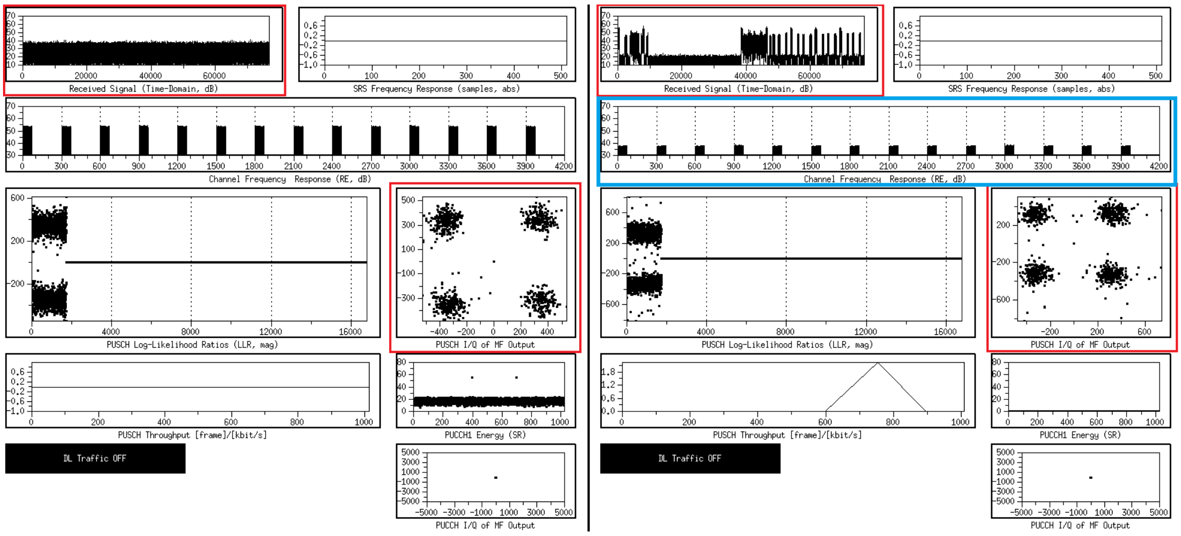

Received Signal (Time-Domain, dB)
SRS Frequency Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
DL Traffic OFF
PUCCH I/Q of MF Output

Figure 6.15: UE running on the *UP Squared*: eNB softscopes for the initial attach state. On the left, FDD band 7. On the right, TDD band 40 (cabled connection).

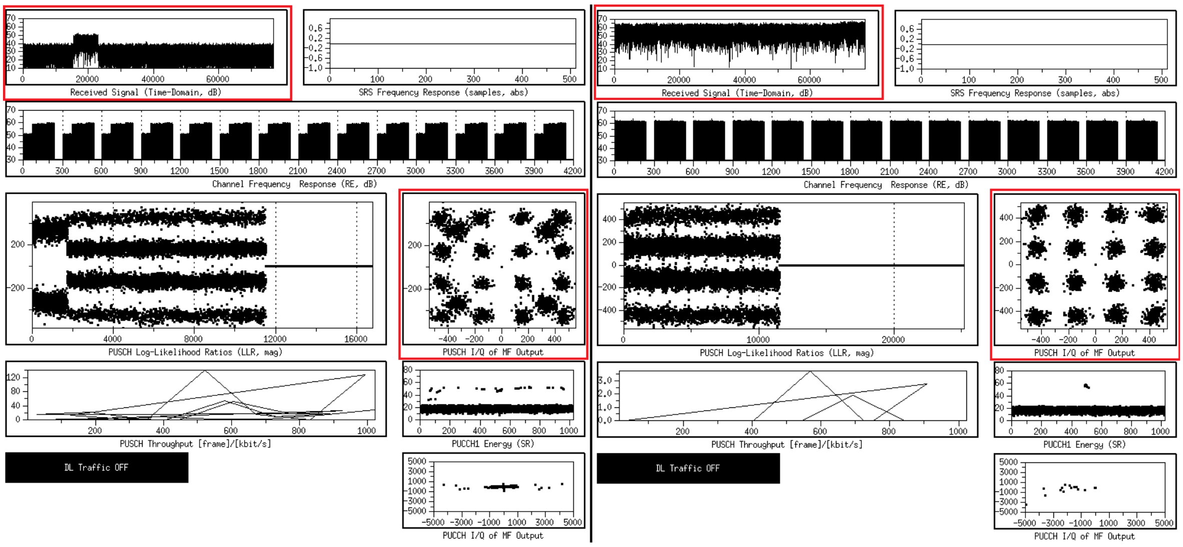

Received Signal (Time-Domain, dB)
SRS Frequency Response (samples, abs)
Channel Frequency Response (RE, dB)
PUSCH Log-Likelihood Ratios (LLR, mag)
PUSCH I/Q of MF Output
PUSCH Throughput [frame]/[kbit/s]
PUCCH1 Energy (SR)
DL Traffic OFF
PUCCH I/Q of MF Output

Figure 6.16: UE running on the *UP Squared*: eNB softscopes for the regular transmission and resource-intensive transmission states, respectively. On the left, FDD band 1 (cabled connection). On the right, FDD band 7.

constellation is a clear representation of a 16-QAM modulation, used in the uplink, in this case during a throughput test.

Three UE implementations were tested: commercial UE, software defined-radio UE running in an *Intel NUC* mini-computer and later in the single-board computer *UP Squared*. Different radio configurations were tested, using the air interface or a cabled connection. Several radio setups were experimented, from using only antennas or a combination between antennas and duplexers to using RF attenuators and a combination of duplexers and RF attenuators.

The LTE FDD bands 1, 5, 7 and 31 and TDD bands 38 and 40 were tested regarding their UDP throughput, latency, PHR and CQI. Table 6.13 presents the successfully tested frequency bands in this dissertation, according to each type of platform.

Table 6.13: Successfully tested frequency bands on the different implementation platforms.

LTE Frequency Band	**Commercial UEs**	**Other UEs (wired)**	**Other UEs (wireless)**
FDD Band 1	✓	✓	✓
FDD Band 5	✓		✓
FDD Band 7	✓		✓
FDD Band 31		✓	
TDD Band 38	✓		
TDD Band 40	✓	✓	

For the commercial UEs, all bands but the FDD band 31 were tested. It is not a regularly used band, so it is not usually recognized by a smartphone.

With a wired connection between the eNB and UE, FDD bands 1 and 31 were tested, along with TDD band 40. As no air interface was utilized, it would be redundant to present results for equal types of radio configurations (e.g. two different FDD frequency bands using a duplexer and attenuator). Although this type of implementation would work in any of the chosen frequency bands, the results were not presented.

For a wireless connection between the eNB and UE, FDD bands 1, 5 and 7 were successfully tested. The other frequency bands would attach to the network but immediately drop the connection. This is mostly caused by radio problems, which can be solved if a duplexing agent is used. In this dissertation's case, a SAW duplexer, which was used in every successful test for the wireless connections setup.

All implementations with a check mark in Table 6.13 allow for Internet browsing, making a Skype call and watching videos on YouTube. This is also true for the free Wi-Fi network created by the software-defined radio UE implementations.

The next and final chapter presents the conclusions of this dissertation's work and explores future work possibilities.

Chapter 7

Conclusions and Future Work

In this chapter, the conclusions are presented in section 7.1 and future work possibilities are explored in section 7.2.

7.1 Conclusions

Through the course of this dissertation, a flexible open-source LTE UE software-defined radio platform was implemented on a compact and low-power SBC device, integrated with an RF hardware front-end from the USRP family. It supports real-time TDD and FDD duplex modes and is reconfigurable for several parameters, like the carrier frequency, E-UTRA bands, bandwidth, number of resource blocks and is able to share its LTE network data with surrounding users, by creating a free Wi-Fi hotspot. First, it was implemented in a mini-computer. Aftwerwards, to turn it into a more portable solution, the UE was ported to a single-board computer, the *UP Squared*.

The LTE standard was introduced regarding its architecture, protocol stack and general physical layers concepts. Afterwards, several experimentation frameworks that implement the LTE protocol were analysed and compared, in order to pick the most adequate for the purpose of this dissertation's work. OpenAirInterface (OAI) was chosen and described.

Then, non-conventional LTE deployment scenarios that could benefit from the flexibility and reconfigurability of the UE implementation were listed. These include catastrophes, support for sporadic events in remote locations, opportunistic use of frequency bands licensed for different purposes, transmission in the unlicensed spectrum, amongst others.

Several network implementations were introduced, focused on different types of user equipments: commercial and defined by software. The processing modules used for each network component were configured, along with the Radio Frequency (RF) front-ends and additionally utilized radio components, such as passive SAW duplexers, antennas and RF attenuators.

The performance of the network was analysed with different implementation setups and was based on measured results of throughput, latency, Power Headroom Report (PHR) and Channel Quality Indicators (CQIs), for different frequency bands and radio configurations. All of the previously mentioned scenarios (in Chapter 4) can make use of the implemented setup.

When comparing the Single-Board Computer (SBC)'s UE deployment results to the ones obtained with the commercial UEs, it is clear that the most optimized is the latter. Still, the results obtained with the single-board computer implementation were better than expected, with it being able to share its LTE mobile data through a Wi-Fi hotspot while maintaining similar throughput values.

7.2 Future Work

As future work, the performance of the SBC could be increased by means of turbo offloading to the boards' FPGA.

Instead of using the Intel x86 architecture, the system should be ported to ARM, as it is much more energy-efficient due to the use of a low-complexity instruction set.

The ability to provide USIM-less authentication should be explored, for easier authentication of foreign users to the network. Currently, a commercial UE would not be able to bypass the authentication, so an app that simulates an USIM card could be created.

The possibility to apply the developed flexible UE implementation to another LTE variants should also be studied, like the LTE-Railway (LTE-R), for high-speed railway communications, or LTE for vehicular communications, to improve the traffic efficiency services and road safety.

Bibliography

[3GP04] 3GPP TR 25.892. Feasibility Study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement (Release 6), June 2004.

[3GP11a] 3GPP TS 36.201. Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description (Release 10), January 2011.

[3GP11b] 3GPP TS 36.322. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification (Release 10), January 2011.

[3GP13] 3GPP TS 36.211. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Release 10), April 2013.

[3GP14a] 3GPP TS 23.003. Numbering, addressing and identification (Release 10), October 2014.

[3GP14b] 3GPP TS 24.301. Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3 (Release 10), October 2014.

[3GP14c] 3GPP TS 36.321. Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification (Release 10), January 2014.

[3GP14d] 3GPP TS 36.323. Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification (Release 10), July 2014.

[3GP15a] 3GPP TS 23.401. General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 10), January 2015.

[3GP15b] 3GPP TS 36.300. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 10), February 2015.

[3GP16a] 3GPP TS 33.401. 3GPP System Architecture Evolution (SAE); Security architecture (Release 10), January 2016.

[3GP16b] 3GPP TS 36.306. Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities (Release 10), January 2016.

[3GP17a] 3GPP TS 36.101. Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (Release 14), April 2017.

[3GP17b] 3GPP TS 36.331. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 10), February 2017.

[Agi09] Agilent Technologies. 3GPP Long Term Evolution: System Overview, Product Development, and Test Challenges. *Application Note*, September 2009.

[Alc09] Alcatel-Lucent. The LTE network Architecture - A comprehensive tutorial. *Strategic White Paper*, 2009.

[ALO12] ALOE++. 2012. Accessed in November 30th, 2017. [Online]. Available in: `https://github.com/agelonch/aloe`.

[Ama17] Amarisoft: Products. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.amarisoft.com/products-lte-ue-ots-sdr-pcie/`.

[Aru11] Arunabha Ghosh, Jun Zhang, Jeffrey G. Andrews, Rias Muhamed. *Fundamentals of LTE*. Pearson Education, Inc., 2011.

[Awa10] Award Solutions. *UE Identifiers in LTE - The Big Five*, 2010. Accessed in November 30th, 2017. [Online]. Available in: `http://lteuniversity.com/get_trained/expert_opinion1/b/dhar/archive/2010/05/31/ue-identities-in-lte-the-big-five.aspx`.

[Cha15] Charles U. Ndujiuba, Oluyinka Oni, Augustus E. Ibhaze. Comparative Analysis of Digital Modulation Techniques in LTE 4G Systems. *Journal of Wireless Networking and Communications*, 2015.

[Chr12] Christopher Cox. *An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications*. John Wiley & Sons, Ltd., 1st edition, 2012.

[Com17] Commel LP-173. 2017. Accessed in November 30th, 2017. [Online]. Available in: `http://www.commell.com.tw/Product/SBC/LP-173.HTM`.

[con17] conga-PA5. 2017. Accessed in November 30th, 2017. [Online]. Available in: `http://www.congatec.com/en/products/pico-itx/conga-pa5.html`.

[Cpu13] Cpuset Management Utility. 2013. Accessed in November 30th, 2017. [Online]. Available in: `https://rt.wiki.kernel.org/index.php/Cpuset_Management_Utility`.

[CPU17] CPU Frequency Scaling. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://wiki.archlinux.org/index.php/CPU_frequency_scaling`.

[Del17] Delock LTE Antenna SMA plug 1 - 4 dBi omnidirectional with tilt joint black. 2017. Accessed in November 30th, 2017. [Online]. Available in: `http://www.delock.com/produkte/G_88451/merkmale.html?setLanguage=en`.

[Eri16] Erik Dahlman, Stefan Parkvall, Johan Sköld. *4G, LTE-Advanced Pro and The Road to 5G*. Elsevier Ltd., 2016.

[Ett17] Ettus Research: Products. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.ettus.com/product/`.

[EUR15] EURECOM. E-UTRAN User Guide, July 2015.

[EUR17a] EURECOM. 2017. Accessed in November 30th, 2017. [Online]. Available in: http://www.eurecom.fr/en.

[EUR17b] EURECOM's GitLab for the OpenAirInterface project. 2017. Accessed in November 30th, 2017. [Online]. Available in: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/home.

[Exp17] ExpressMIMO2 4G Board. 2017. Accessed in November 30th, 2017. [Online]. Available in: http://www.expressmimo2.com/.

[Fle15] Flexicell OAI-based Cloud Radio Access Networks. 2015. Accessed in November 30th, 2017. [Online]. Available in: http://www.openairinterface.org/?page_id=1638.

[Flo17] Florian Kaltenberger. OAI Workshop: OpenAirInterface 5G Overview, Installation, Usage, April 2017.

[fre11] freeDiameter: Diameter open implementation. 2011. Accessed in November 30th, 2017. [Online]. Available in: http://www.freediameter.net/trac/.

[gr-13] gr-LTE: GNU Radio LTE Receiver. 2013. Accessed in November 30th, 2017. [Online]. Available in: https://github.com/kit-cel/gr-lte/.

[Hac17] HackRF One. 2017. Accessed in November 30th, 2017. [Online]. Available in: http://greatscottgadgets.com/hackrf/.

[Hen16] Hengyang Shen, Xingguang Wei, Haitao Liu, Yang Liu, Kan Zheng. Design and implementation of an LTE system with multi-thread parallel processing on OpenAirInterface platform. *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th*, pages 1–5, 2016.

[Int17] Intel Joule 550x Developer Kit with Expansion Board. 2017. Accessed in November 30th, 2017. [Online]. Available in: https://pt.mouser.com/new/Intel/intel-joule-550x-dev-kit/.

[ipe17] iperf3. 2017. Accessed in November 30th, 2017. [Online]. Available in: http://software.es.net/iperf/.

[Ism16] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano, Douglas J. Leith. srsLTE: An Open-Source Platform for LTE Evolution and Experimentation. *WiNTECH@MobiCom*, 2016.

[Key17] Keysight 85052D Economy Mechanical Calibration Kit, DC to 26.5 GHz, 3.5 mm. 2017. Accessed in November 30th, 2017. [Online]. Available in: http://www.keysight.com/en/pd-1000002018%3Aepsg%3Apro-pn-85052D/economy-mechanical-calibration-kit-dc-to-265-ghz-35-mm?cc=US&lc=eng.

[Lim17] Lime Microsystems: Software Define Radio. 2017. Accessed in November 30th, 2017. [Online]. Available in: http://www.limemicro.com/products/software-defined-radio/.

[LMRZ17] Mina Labib, Vuk Marojevic, Jeffrey H Reed, and Amir I Zaghloul. Extending lte into the unlicensed spectrum: technical analysis of the proposed variants. *arXiv preprint arXiv:1709.04458*, 2017.

[LSWW14] Ming-Feng Lee, Nigel P Smart, Bogdan Warinschi, and Gaven J Watson. Anonymity guarantees of the umts/lte authentication and connection protocol. *International journal of information security*, 13(6):513–527, 2014.

[Mar16] Markus Ahlström, Simon Holmberg. Prototype Implementation of a 5G Group-Based Authentication and Key Agreement Protocol. Master's thesis, Lund University, Sweden, 2016.

[Mor08] Moray Rumney. 3GPP LTE: Introducing Single-Carrier FDMA. *Agilent Measurement Journal*, January 2008.

[Net17] Network Cell Info Lite app. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://play.google.com/store/apps/details?id=com.wilysis.cellinfolite&hl=en`.

[ns-17] ns-3 LTE Module. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.nsnam.org/docs/models/html/lte.html`.

[Nua17] Nuand bladeRF. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.nuand.com/`.

[nwE12] nwEPC: SAE/EPC Serving Gateway and Packed Data Network Gateway. 2012. Accessed in November 30th, 2017. [Online]. Available in: `https://github.com/thomasbhatia/nwEPC---EPC-SAE-Gateway`.

[One17] OnePlus 3. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://oneplus.net/pt/3`.

[Ope13] Open Source Long-Term Evolution (LTE) Deployment. 2013. Accessed in November 30th, 2017. [Online]. Available in: `https://sites.google.com/site/osldproject/`.

[Ope17a] OpenAirInterface. 2017. Accessed in November 30th, 2017. [Online]. Available in: `http://www.openairinterface.org/`.

[Ope17b] OpenEPC. 2017. Accessed in November 30th, 2017. [Online]. Available in: `http://www.openepc.com/`.

[ope17c] openLTE. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://sourceforge.net/projects/openlte/`.

[Pow14] Power Management States: P-States, C-States, and Package C-States. 2014. Accessed in November 30th, 2017. [Online]. Available in: `https://software.intel.com/en-us/articles/power-management-states-p-states-c-states-and-package-c-states`.

[Qua09] Qualcomm RF360 Europe GmbH. Application Note. SAW-Components: How to choose the optimal SAW filter, January 2009.

[Qua10] Qualcomm RF360 Europe GmbH. Application Note SAW components: Layout considerations for EPCOS SAW filters, March 2010.

[Qua16a] Qualcomm RF360 Europe GmbH. BAW/SAW duplexer for small cell, LTE band 7, Series/type: B8032, March 2016.

[Qua16b] Qualcomm RF360 Europe GmbH. SAW duplexer, LTE band 1, Series/type B8651, May 2016.

[Qua17] Qualcomm RF360 Europe GmbH. SAW duplexer Small cell and femtocell, LTE band 5, Series/type: B8013, May 2017.

[RO417] RO4360G2 Laminates. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.rogerscorp.com/acs/products/56/RO4360G2-Laminates.aspx`.

[RTL17] RTL-SDR. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.rtl-sdr.com/`.

[Sam17] Samsung Galaxy S4 Active. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.samsung.com/us/mobile/phones/galaxy-s/samsung-galaxy-s4-active-at-t-gray-sgh-i537zaaatt/#specs`.

[Spe17] Speedtest.net by Ookla. 2017. Accessed in November 30th, 2017. [Online]. Available in: `www.speedtest.net/`.

[SRS17] SRS: Products. 2017. Accessed in November 30th, 2017. [Online]. Available in: `http://www.softwareradiosystems.com/products/`.

[Ste11] Stefania Sesia, Issam Toufik, Matthew Baker. *LTE - The UMTS Long Term Evolution: From Theory to Practice.* John Wiley & Sons, Ltd., 2nd edition, 2011.

[Tin17] Tinkercad. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.tinkercad.com`.

[Tor10] Tore Ulversoy. Software Defined Radio: Challenges and Opportunities. *IEEE Communications Surveys & Tutorials*, 12(4):531–550, 2010.

[UDO17] UDOO x86. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.udoo.org/udoo-x86/`.

[Up 17a] Up Squared. UP Squared Specification/Datasheet, 2017.

[UP 17b] UP Squared Pentium Quad Core 8GB memory/64GB eMMC. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://up-shop.org/up-boards/97-up-squared-pentium-quad-core-8gb-memory64gb-emmc.html`.

[Wir17] Wireshark. 2017. Accessed in November 30th, 2017. [Online]. Available in: `https://www.wireshark.org/`.

[Yan10] Samuel C Yang. *OFDMA system analysis and design.* Artech House, 2010.

Appendix A

Authentication

One of this dissertation's interests was to try and skip the network authentication process, so any commercial UE could connect to a free and flexible network. It was proven that such a scenario was not possible and the reasons why are explained in this Annex. First, LTE's mutual authentication procedure is explained. Then, the several types of attach are discussed, with focus on the emergency attach, which allows the user to attach to the network without any security functionalities. The ciphering and integrity protection algorithms are also described.

Some of the LTE security functions include [3GP15a]:

- Guards against unauthorised EPS service usage (authentication of the UE by the network and service request validation).
- Provision of user identity confidentiality (temporary identification and ciphering).
- Provision of user data and signalling confidentiality (ciphering).
- Provision of origin authentication of signalling data (integrity protection).
- Authentication of the network by the UE.

From the simplified authentication call flow, in Figure A.1, the main authentication steps can be identified. First, a UE communicates with the EPC in order to start the procedure. It then passes through the EPS AKA protocol, where common integrity and ciphering keys are established. Afterwards, the network chooses the highest preference integrity and encryption algorithms according to the UE's capabilities and integrity and cyphering is initiated.

The security keys are verified using the integrity algorithm and IK and if the received message authentication code is valid, the attach is now established and both parties have been authenticated. Every exchanged message is now protected by the defined CK and IK keys and security algorithms.

The EPS AKA protocol is one of the reasons why the network authentication can't be bypassed and will be described in the following section.

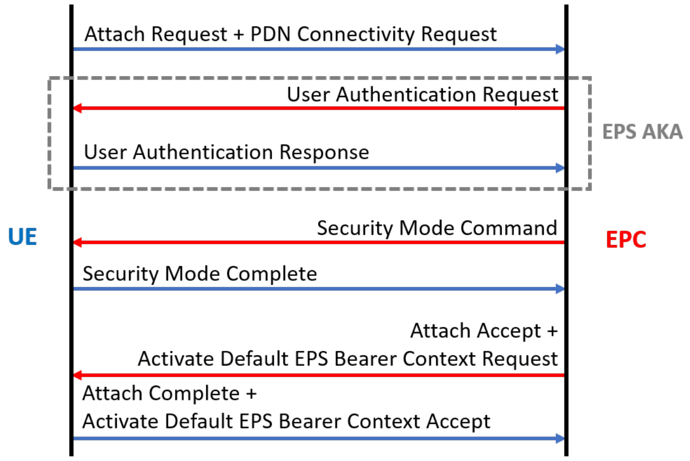

Attach Request + PDN Connectivity Request
User Authentication Request
EPS AKA
User Authentication Response
UE
Security Mode Command
EPC
Security Mode Complete
Attach Accept +
Activate Default EPS Bearer Context Request
Attach Complete +
Activate Default EPS Bearer Context Accept

Figure A.1: Simplified authentication call flow between the UE and EPC.

A.1 EPS Authentication and Key Agreement (AKA)

EPS AKA is the mutual authentication and key agreement procedure that is used over E-UTRAN, between the UE and MME.

Between the HSS and the UE, three keys are shared: K, CK and IK. K is permanently stored on the USIM card and in the AuC. CK and IK are derived in the AuC and on the USIM during an AKA run [3GP16a]. Figure A.2 provides a detailed overview of the EPS AKA protocol. The sequence number SQN is a counter that supports network authentication, existent in both UE and network but maintained separately. The initial values for SQN are set to zero and the two counters are incremented with each authentication. They are used to assure the freshness of the protocol.

The AKA protocol relies on message authentication functions (f1 and f2) and key generation functions (f3, f4 and f5), all of which controlled by the same key (K). As shown in Figure A.2, this key is extracted from the HSS by using the UE's IMSI. Afterwards, individual Authentication Vectors (AVs) are generated in the network and the process goes as follows: first, a random number RAND is generated and a fresh sequence number SQN is obtained. Then, a message authentication code (MAC), an Expected Response (XRES), an Anonimity Key (AK), a Confidentiality Key (CK) and an Integrity Key (IK) are calculated, as in Table A.1. The AMF is used for operator-specific options, e.g. its bit 0 defines if the attach is for E-UTRAN or for previous generations. The AK is used to provide anonimity to the SQN, but can also be set to 0. Afterwards, an Authentication Token (AUTN) and AV are created.

Before generating the ciphering and integrity keys, a second procedure must be completed. After a fresh AV is generated, the random challenge and authentication token (RAND and AUTN) are sent to the UE in an Authentication Request message. Upon receipt of these values, the UE computes the AK to retrieve the sequence number (SQN=(SQN⊕AK)⊕AK) from the authentication token. The XMAC is then computed and compared with the network MAC. If they are equal, the process can continue. The calculated SQN must always be compared with the network's SQN. Again, if they are in a correct range, the process can go on and the UE can calculate a Response (RES), which is sent back to the network in an Authentication Response message. The UE computes can finally computer the CK and IK.

To finalize the AKA protocol, the network compares the received response RES with the

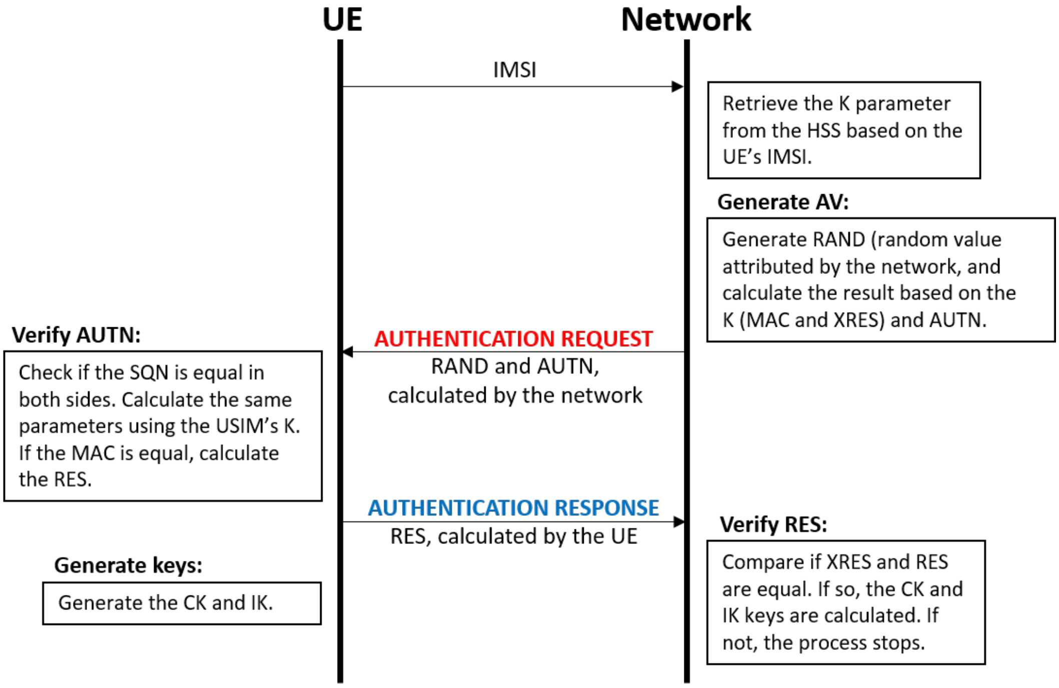

UE
Network
IMSI
Retrieve the K parameter from the HSS based on the UE's IMSI.
Generate AV:
Generate RAND (random value attributed by the network, and calculate the result based on the K (MAC and XRES) and AUTN.
Verify AUTN:
Check if the SQN is equal in both sides. Calculate the same parameters using the USIM's K. If the MAC is equal, calculate the RES.
AUTHENTICATION REQUEST
RAND and AUTN, calculated by the network
AUTHENTICATION RESPONSE
RES, calculated by the UE
Verify RES:
Compare if XRES and RES are equal. If so, the CK and IK keys are calculated. If not, the process stops.
Generate keys:
Generate the CK and IK.

Figure A.2: EPS Authentication and Key Agreement (AKA).

expected response XRES from the AV. If they are the same, the UE passes the authentication and the CK/IK keys are extracted. If not, the network rejects the authentication procedure [LSWW14].

With this information, it's clear that the K is essential for LTE authentication, and a UE from a different network operator can't simply connect to a network of its choice without both knowing this value. Another approach has to be taken.

A.2 Attach types and Security Algorithms

The Attach Request message includes IEs such as EPS attach type, EPS mobile identity, UE network capability, ESM message container, Additional GUTI, Last visited registered TAI and UE's usage settings. The IE EPS Attach Type indicates the purpose of the attach procedure. There are three main types of attach: EPS attach (for EPS services only), combined EPS/IMSI attach (for both EPS and non-EPS services) and EPS emergency attach (for emergency bearer services).

Each type of attach supports different ciphering and integrity protection algorithms, EPS Encryption Algorithm (EEA) and EPS Integrity Algorithm (EIA) (described in Table A.2). These algorithms have a 128-bit input key except for the null ciphering and integrity protection algorithms.

The 128-EEA1/EIA1 algorithms are based in SNOW 3G, a stream cipher, which allows for low power consumption. The 128-EEA2/EIA2 algorithms are based in AES, a block cipher, and are as different from SNOW 3G as possible (by cracking one method, the other would

Table A.1: EPS AKA parameters derivation.

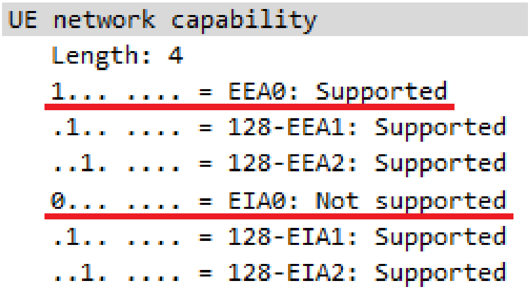
Parameters	Calculation
K	Pre-configured in HSS and USIM
OP	Pre-configured in HSS and USIM
AMF	Pre-configured in HSS and USIM
SQN	Sequence refreshed in each authentication
RAND	Randomly generated
MAC / XMAC	$f1_K(\text{MAC}\Vert\text{RAND}\Vert\text{AMF})$
RES / XRES	$f2_K(\text{RAND})$
CK	$f3_K(\text{RAND})$
IK	$f4_K(\text{RAND})$
AK	$f5_K(\text{RAND})$
AUTN	$(\text{SQN}\oplus\text{AK})\Vert\text{AMF}\Vert\text{MAC}$
AV	$\text{RAND}\Vert\text{XRES}\Vert\text{CK}\Vert\text{IK}\Vert\text{AUTN}$

Table A.2: Ciphering and integrity protection algorithms.

Value	Ciphering	Integrity
0001	EEA0 (Null Ciphering)	EIA0 (Null Integrity)
0010	EEA1 (SNOW 3G)	EIA1 (SNOW 3G based)
0100	EEA2 (AES)	EIA2 (AES based)

not be affected).

The EEA0/EIA0 algorithms must only be used for the EPS emergency attach, which is the same as having no security algorithms. The implementation of the EIA0 in MMEs and eNBs is optional and, if implemented, shall be disabled in MMEs and eNBs where support of unauthenticated emergency calling is not required. Most UEs don't support the EIA0 algorithm as well, such as the ones used for testing in this dissertation, the Samsung Galaxy S4 and OnePlus 3. This can be found in the Wireshark trace for the Galaxy S4's UE network capability during the Attach Request message, in Figure A.3.



```
UE network capability
    Length: 4
    1... .... = EEA0: Supported
    .1.. .... = 128-EEA1: Supported
    ..1. .... = 128-EEA2: Supported
    0... .... = EIA0: Not supported
    .1.. .... = 128-EIA1: Supported
    ..1. .... = 128-EIA2: Supported
```

Figure A.3: Null integrity protection algorithm unsupported by Samsung Galaxy S4.

For the EPS and combined EPS/IMSI attaches, when using the OAI MME one can force the null algorithms EEA0/EIA0 on the network side but, when the UE attaches to the network, it will end up using the EEA0/EIA1 algorithms. This happens because the user can't select the algorithm the commercial UE is going to use and it is programmed to only use the EIA0 integrity protection in case of an EPS emergency attach.

This means that the only alternative to surpass the mutual authentication process and ciphering and integrity protection algorithms would be to force the UE to use its EPS emergency attach to connect to the network. However, it is not, and the reason why will be described in the next section.

A.3 Emergency Attach

The problem with the emergency attach is that is it only used to establish IMS calls, through the means of EPS emergency bearer services.

Emergency bearer services support IMS emergency sessions and are provided by the serving network when it is configured to support emergency services. Emergency bearer services are provided to normal attached or emergency attached UEs and depending on local regulation, to UEs that are in limited service state. Receiving emergency services in limited service state does not require a subscription. Four different levels of emergency bearer support are described below:

- **Valid UEs only:** only for UEs that have a valid subscription and are authenticated (limited service state is not allowed). UEs should be attached to the network and then perform a PDN Connection Request when an IMS emergency session is detected by the UE.

- **Authenticated UEs only:** these UEs must have a valid IMSI. These UEs are authenticated and may be in limited service state due to being in a location that they are restricted from service. A UE that can not be authenticated will be rejected.

- **IMSI required, but the authentication is optional:** these UEs must have an IMSI. If authentication fails, the UE is granted access and the unauthenticated IMSI retained in the network for recording purposes. The IMEI is used in the network as the UE identifier.

- **All UEs are allowed:** along with authenticated UEs, this includes UEs with an IMSI that can not be authenticated and UEs with only an IMEI. If an unauthenticated IMSI is provided by the UE, the unauthenticated IMSI is retained in the network for recording purposes. The IMEI is used in the network to identify the UE.

To provide emergency bearer services, the MME is configured with MME Emergency Configuration Data that are applied to all emergency bearer services that are established by an MME on UE request. The MME Emergency Configuration Data contain the Emergency APN which is used to derive a PDN GW, or the MME Emergency Configuration Data may also contain the statically configured PDN GW for the Emergency APN. The network supporting emergency services for UEs in limited service state provides emergency bearer services to these UE, regardless whether the UE can be authenticated, has roaming or mobility restrictions or a valid subscription.

For a UE that is Emergency Attached, if it is unauthenticated the EPS security context is not set up on UE. Normal attached UEs initiate the UE Requested PDN Connectivity procedure to receive emergency bearer services [3GP15a].

As there's no access to mobile data during the Emergency Attach and it is only used for IMS emergency sessions, when the call is finished the user would have to go through the normal attach procedure to have access to the regular LTE network. So, although no security is enforced during the Emergency Attach and all UEs are allowed emergency bearers, it is not useful to force a UE to go through it to bypass the mutual authentication and encryption procedures, as there would never be access to LTE mobile data, for example.

The authentication procedure must always be completed, so this idea will be discarded, and another approach will be taken. Instead of having a free LTE network that every user could connect to, one abstracted UE will connect to the LTE network and then provide its LTE mobile data with the surrounding devices, by means of a free Wi-Fi hotspot.

Appendix B

Passive Surface Acoustic Wave (SAW) Duplexers Design

In the context of this dissertation, passive SAW duplexers for three different LTE FDD bands were designed and built, in order to test different band operation, prove the flexibility of the device and improve the signal quality.

B.1 Board Design

Different duplexer chips for LTE small-cell from Qualcomm were used, for the frequency bands 1 [Qua16b], 5 [Qua17] and 7 [Qua16a]. All duplexer packages are SMT, so they are very small. Because of this, their power-handling capabilities are not so great, meaning they only work for low-power transmission.

The influence of the PCB layout gets stronger with higher frequencies. Too long lines or too thin lines between the SAW filter and the additional components will add additional losses. The holes should be as close to the pad as possible, or the matching results may differ, and there should be two holes per pad, not smaller than 0.2 mm [Qua10].

In accordance to the frequencies being used, the laminate for the duplexers' board was chosen: Rogers RO4360G2, with thickness 0.81mm and a DK (dielectric constant) of 6.15. This laminate's typical applications are power amplifiers, patch antennas and other RF components, such as couplers or diplexers [RO417]. The copper wire's thickness is 1.17 mm and the holes have a radius of 0.3 mm (only one per pad). The length of the copper wire from the inductors to the SMA connectors is 7 mm and the board sizes are approximately 2x1.5 mm. The duplexer ground pads were merged with copper lines and extended to the sides of the chip, so holes could be drilled.

The external matching circuits recommended in the datasheet of the components are different for each bands, so they will be discussed in separate sections.

B.1.1 Band 1

This low-loss SAW duplexer for LTE Band 1 provides usable pass bands of 60 MHz. This means 2110-2170 MHz for the uplink (RX) and 1920-1980 MHz for the downlink (TX).

The external matching circuit only requires an inductor connected to the antenna port. Figure B.1 shows the matching circuit schematic and board design.

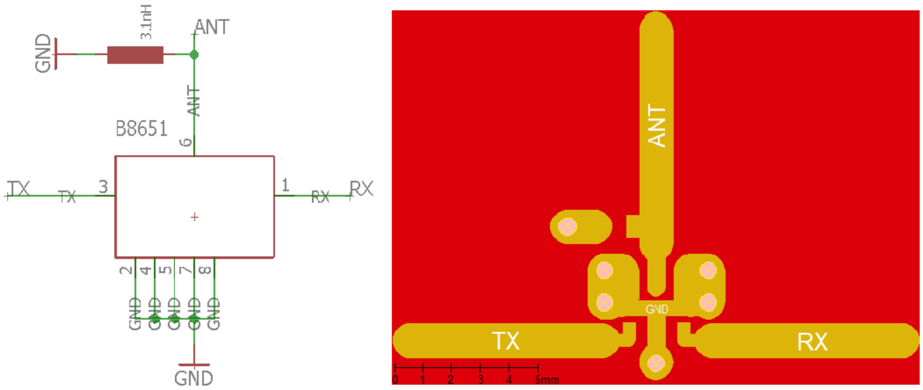


Figure B.1: Band 1 duplexer's matching circuit schematic and board design.

B.1.2 Band 5

The LTE Band 5 low-loss SAW duplexer provides usable pass bands of 25 MHz, i.e., 824-849 MHz for the uplink (RX) and 869-894 MHz for the downlink (TX).

No external matching is required. Figure B.2 shows the Band 5 duplexer's schematic and board design.

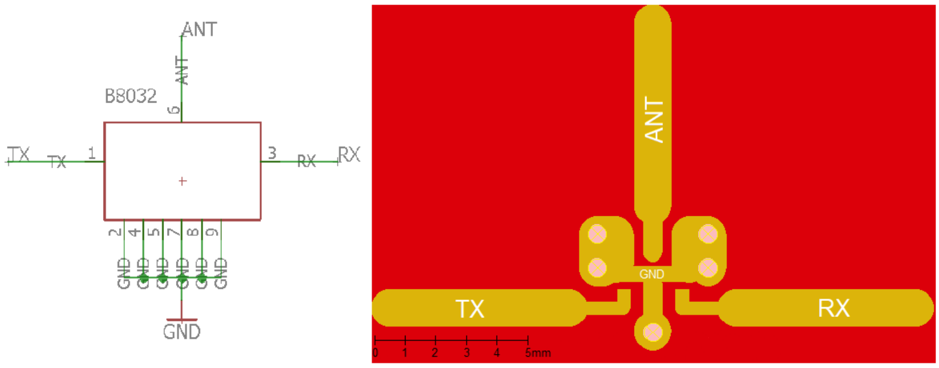


Figure B.2: Band 5 duplexer's matching circuit schematic and board design.

B.1.3 Band 7

The low-loss LTE Band 7 SAW duplexer is a duplexer for small cell systems, with 70 MHz of usable pass band: 2500-2570 MHz on the RX port and 2620-2690 MHz on the TX.

It requires three matching inductors, one for each port. Figure B.3 shows the matching circuit schematic and board design.

The inductors used for the matching circuits are SMD components and have a package size 0603. The final step is, by using a CAM[2] processor, to extract the gerber files of the top copper layer, bottom copper layer (which will serve as the outline for the board itself) and

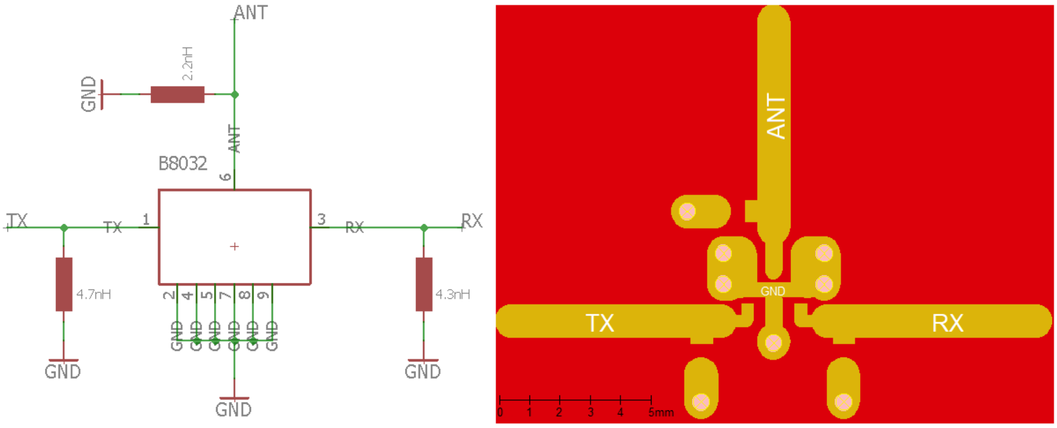


Figure B.3: Band 7 duplexer's matching circuit schematic and board design.

drills/holes file. They are now ready for the CNC[3] router machine, a computer-controlled cutting machine.

B.2 S-parameters

In order to measure if the attenuation level on the duplexers was working as intended, the 4-port S-parameters were measured for each individual board. The used machine was a PNA-X Network Analyzer, from Agilent Technologies.

Because 4-port S-parameters are being measured, Port 1 will connect to the TX port of the duplexer, Port 3 to the antenna port and Port 4 to the RX port. Port 2 is not relevant. Before measuring, a calibration has to be ensued, using the mechanical calibration kit 85052D 3.5mm [Key17].

After calibration, the duplexers can be measured individually. The S-parameters for checking the attenuation levels are the S13 (TX-ANT) and S43 (ANT-RX). The bands closer to the 0 dB attenuation are the passbands, the rest should be highly attenuated. After measuring, the S-parameters matrix is extracted and then the S13 and S43 parameters are plotted in MATLAB, using the `rfplot` function.

This `rfplot` functions takes the extracted scattering matrix and plots whichever parameters the user desires. The plots for each frequency band will be presented in the following sections.

B.2.1 Band 1

Figure B.4, the S-parameters between the TX-ANT (S13) ports and ANT-RX (S43) ports are displayed.

The signal attenuation is around -2 dB in the duplexers' passbands (1920-1980 MHz and 2100-2170 MHz). In those bands it is quite flat, except for a small ripple on the DL's frequency, so the circuit is well matched. Also, the frequencies outside the passbands are well attenuated, meaning the selectivity of the Band 1 SAW duplexer is very high.

[2] Computer-Aided Manufacturing (CAM)

[3] Computer Numerical Control (CNC)

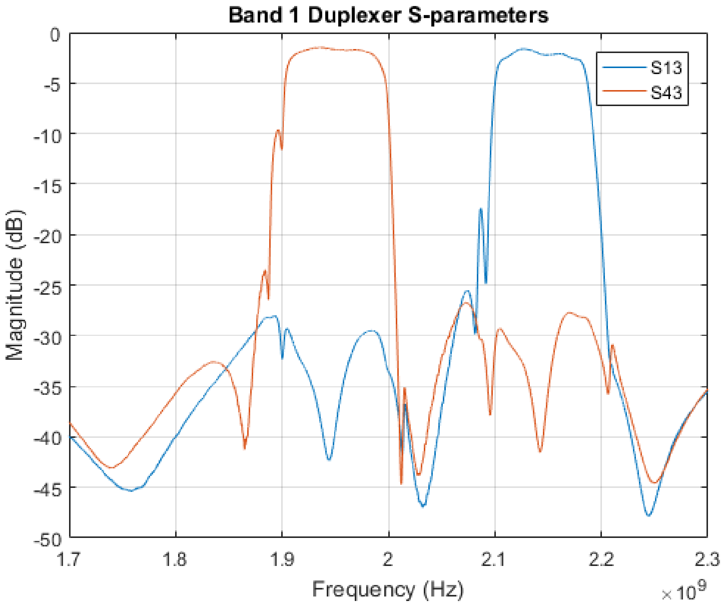


Figure B.4: Band 1 duplexer S-parameters.

B.2.2 Band 5

Figure B.5, the S-parameters between the TX-ANT (S13) ports and ANT-RX (S43) ports are displayed.

This duplexer's attenuation in the passband is not flat, there's a lot of ripple from -2 to -5 dB. The usable passband goes from 830-855 MHz on the UL and 870-905 MHz on the DL. Every other frequency is very well attenuated. As the usable passband is not consistent, the frequency bandwidth of the signal shouldn't be higher than 5 MHz (25 RBs) when testing.

B.2.3 Band 7

In Figure B.6, the S-parameters between the TX-ANT (S13) ports and ANT-RX (S43) ports are displayed.

The signal attenuation is around -2 dB in the duplexers' passbands and it is flat throughout the whole passband, meaning the circuit is well matched. The usable bands are 2490-2580 MHz for the UL and 2620-2710 MHz for the DL. All other frequencies are attenuated, so the selectivity is high.

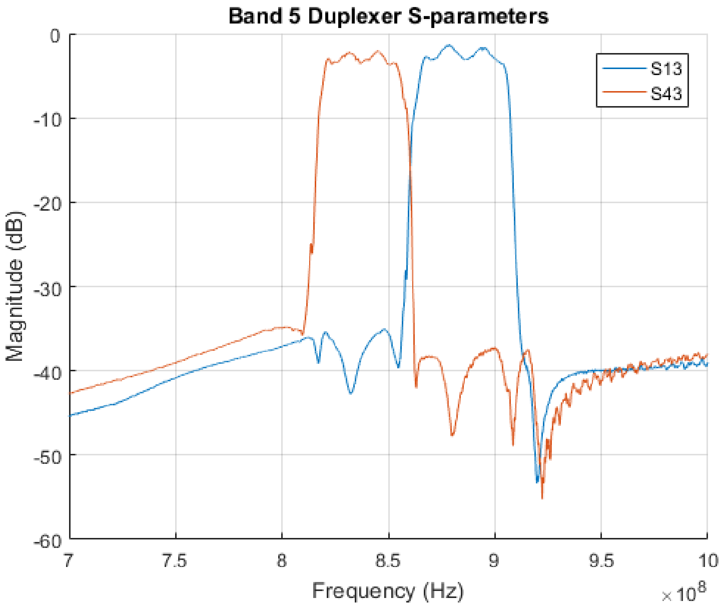


Figure B.5: Band 5 duplexer S-parameters.

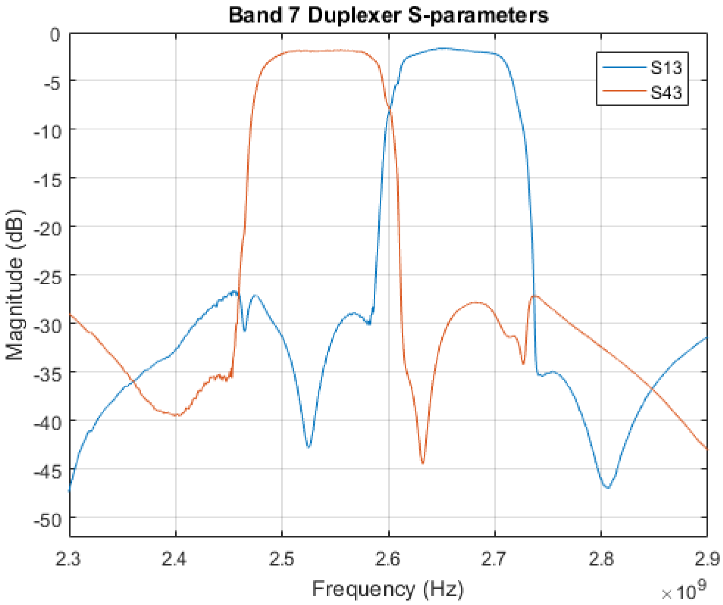


Figure B.6: Band 7 duplexer S-parameters.

Appendix C

OpenAirInterface (OAI) Installation and Use Guide

This appendix provides a guide on how to install, configure and execute OAI's eNB and EPC connected with a UE (commercial or OAI's), on Ubuntu 16.04.3 LTS 64-bit machines connected with commercial RF front-ends. The UE traffic is routed to the Internet.

Three computers are configured, one for each network component (eNB, EPC, UE); then, the UE is migrated to a SBC. Each computer's specifications are presented in Chapter 5 of this dissertation. A new network is created, with a different PLMN-ID (26808) from the regular network operators. Several ways on how to use the abstracted UE to provide Internet access to surrounding devices are also described.

All information in this guide was compiled from the OAI's website tutorials, mailing lists, documents and based on personal experience with the platform.

C.1 Installation

In order to correctly prepare the three computers for running the OAI's software, the following steps have to be performed.

- Install Ubuntu 16.04.3 LTS (64-bit).
- Disable C-states, turbo mode, hyper-threading, Intel SpeedStep technology and other power management tools in BIOS (this step can be skipped for the EPC's computer).
- Disable P-states/C-States in the Linux boot options, by adding the following line in `/etc/default/grub`, and then updating the grub (this step can be skipped for the EPC's computer).

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_pstate=disable
    processor.max_cstate=1 intel_idle.max_cstate=0 idle=poll"
update-grub
```

- Disable CPU frequency scaling (this step can be skipped for the EPC's computer).

```
sudo apt install cpufrequtils
sudo nano /etc/init.d/cpufrequtils
```

Afterwards, add the following line to it, and disable the `ondemand` daemon, otherwise the settings will be overwritten on reboot.

```
GOVERNOR="performance"
sudo update-rc.d ondemand disable
```

- Append ”blacklist intel_powerclamp” to the end of `/etc/modprobe.d/blacklist.conf` (this step can be skipped for the EPC's computer).
- Install git and add the OAI repository as an authorized remote system:

```
sudo apt install git
echo -n | openssl s_client -showcerts -connect gitlab.eurecom.fr:443
    2>/dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' |
    sudo tee -a /etc/ssl/certs/ca-certificates.crt
```

- Install a low-latency kernel (above 4.8 for the RAN and 4.7 for the EPC):
  - eNB/UE:

```
sudo apt install linux-image-4.8.0-46-lowlatency
    linux-headers-4.8.0-46-lowlatency
```

  - EPC (this one already contains the GTP module in a pre-compiled Debian package):

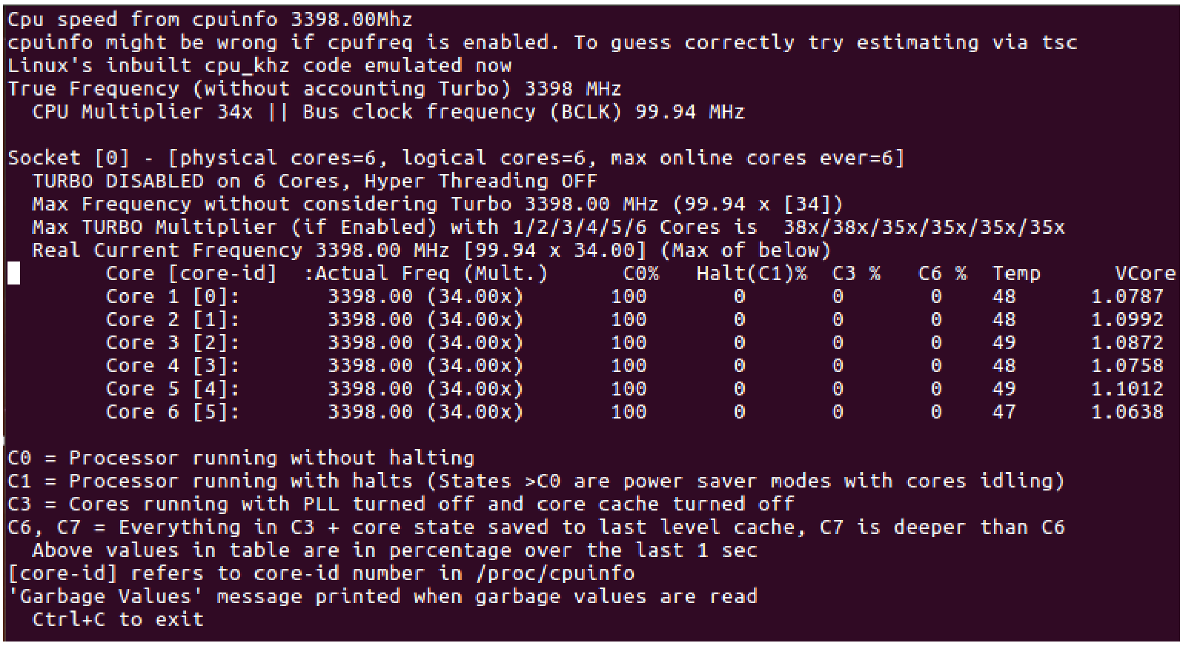
```
git clone https://gitlab.eurecom.fr/oai/linux-4.7.x.git
cd linux-4.7.x
sudo dpkg -i
    linux-headers-4.7.7-oaiepc_4.7.7-oaiepc-10.00.Custom_amd64.deb
    linux-image-4.7.7-oaiepc_4.7.7-oaiepc-10.00.Custom_amd64.deb
```

- Install the i7z tool, to check the CPU speed. Every core should be at the same frequency, with hyper-threading and turbo turned off (Figure C.1). The only active C-state should be C0, indicating the processor never goes into idle mode (this step can be skipped for the EPC's computer).
- Specify a fully qualified domain name (FQDN) for each computer. Modify /etc/hosts, replacing `flexicell3` with the computer's hostname. It should look like this:

```
127.0.0.1  localhost
127.0.1.1  flexicell3.openair5G.eur flexicell3
127.0.1.1  hss.openair5G.eur hss
```

- Get the OAI software from their GitLab server and checkout the develop branch:
  - RAN (eNB/UE):

```
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
```



```
Cpu speed from cpuinfo 3398.00Mhz
cpuinfo might be wrong if cpufreq is enabled. To guess correctly try estimating via tsc
Linux's inbuilt cpu_khz code emulated now
True Frequency (without accounting Turbo) 3398 MHz
  CPU Multiplier 34x || Bus clock frequency (BCLK) 99.94 MHz

Socket [0] - [physical cores=6, logical cores=6, max online cores ever=6]
  TURBO DISABLED on 6 Cores, Hyper Threading OFF
  Max Frequency without considering Turbo 3398.00 MHz (99.94 x [34])
  Max TURBO Multiplier (if Enabled) with 1/2/3/4/5/6 Cores is  38x/38x/35x/35x/35x/35x
  Real Current Frequency 3398.00 MHz [99.94 x 34.00] (Max of below)
        Core [core-id]  :Actual Freq (Mult.)      C0%   Halt(C1)%  C3 %   C6 %  Temp      VCore
        Core 1 [0]:       3398.00 (34.00x)       100       0       0       0    48      1.0787
        Core 2 [1]:       3398.00 (34.00x)       100       0       0       0    48      1.0992
        Core 3 [2]:       3398.00 (34.00x)       100       0       0       0    49      1.0872
        Core 4 [3]:       3398.00 (34.00x)       100       0       0       0    48      1.0758
        Core 5 [4]:       3398.00 (34.00x)       100       0       0       0    49      1.1012
        Core 6 [5]:       3398.00 (34.00x)       100       0       0       0    47      1.0638

C0 = Processor running without halting
C1 = Processor running with halts (States >C0 are power saver modes with cores idling)
C3 = Cores running with PLL turned off and core cache turned off
C6, C7 = Everything in C3 + core state saved to last level cache, C7 is deeper than C6
  Above values in table are in percentage over the last 1 sec
[core-id] refers to core-id number in /proc/cpuinfo
'Garbage Values' message printed when garbage values are read
  Ctrl+C to exit
```

Figure C.1: i7z output.

- EPC:

  ```
  git clone https://gitlab.eurecom.fr/oai/openair-cn.git
  ```

- Checkout the branch with the most recent commits, develop.

  ```
  git checkout develop && git pull
  ```

The three computers are now ready to run the software, so each will be configured according to their specific function.

C.2 eNB's computer

- Check if the 4.8 low-latency kernel is installed. `uname -a` should output:

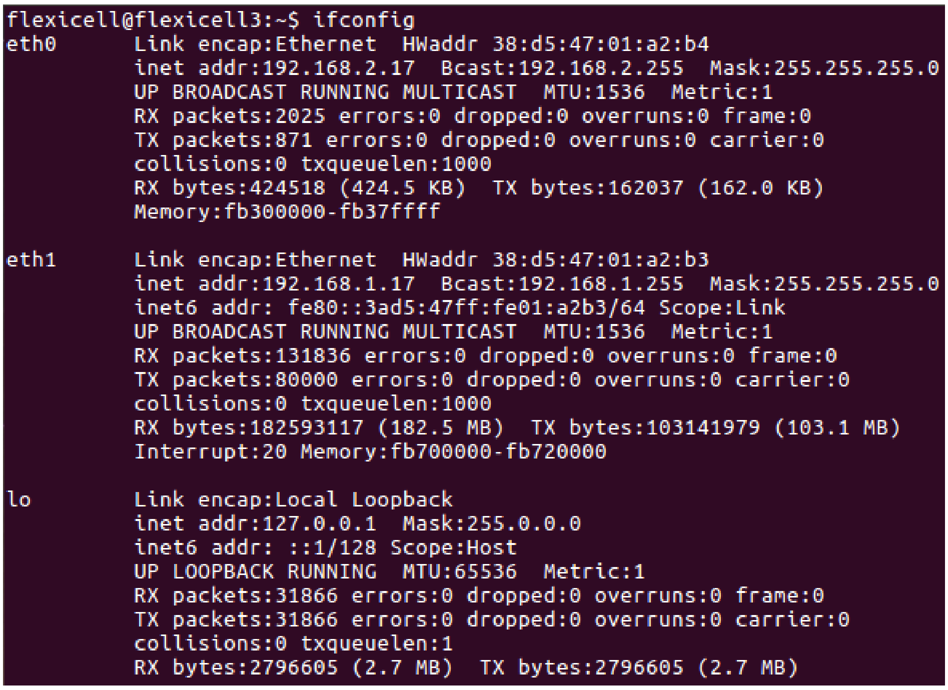
  ```
  uname -a

    Linux flexicell3 4.8.0-46-lowlatency #49~16.04.1-Ubuntu SMP PREEMPT Fri Mar
        31 16:31:01 UTC 2017 x86_64 x86_64
  ```

- Confirm the develop branch is the one in use (indicated by the asterisk).

  ```
  cd openairinterface5g && git branch -vvv

    * develop 38abafe [origin/develop] Merge branch 'develop\_integration\_w14'
        into 'develop'
  ```

```
    master 82e5410 [origin/master: behind 1] add hotfix to add usrp ud-host
        package
```

- Configure the network interface, in this case eth1, with a manual IP address (e.g. 192.168.1.17), as in Figure C.2.



```
flexicell@flexicell3:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 38:d5:47:01:a2:b4
          inet addr:192.168.2.17  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1536  Metric:1
          RX packets:2025 errors:0 dropped:0 overruns:0 frame:0
          TX packets:871 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:424518 (424.5 KB)  TX bytes:162037 (162.0 KB)
          Memory:fb300000-fb37ffff

eth1      Link encap:Ethernet  HWaddr 38:d5:47:01:a2:b3
          inet addr:192.168.1.17  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::3ad5:47ff:fe01:a2b3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1536  Metric:1
          RX packets:131836 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80000 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:182593117 (182.5 MB)  TX bytes:103141979 (103.1 MB)
          Interrupt:20 Memory:fb700000-fb720000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:31866 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31866 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:2796605 (2.7 MB)  TX bytes:2796605 (2.7 MB)
```

Figure C.2: eNB's computer `ifconfig` Linux command.

- Using OAI's build wrapper, in openairinterface5g's cmake_targets folder, the eNB is built and additional required software packages are installed.

```
cd openairinterface5g/
source oaienv
./cmake_targets/build_oai -I --eNB -w USRP -c -C
```

Summary of the used flags:

  - -I: installs additional required software packages, such as freeDiameter. It's only needed in the first execution.
  - --eNB: compiles eNBs' lte-softmodem.
  - -w: selects the hardware target and adds platform support: USRP, in this case.
  - -c: erases all files to make a rebuild from start.
  - -C: erases all files made by previous compilations and installations.

- Add the new network's MCC (268) and MNC (08) to the list of mobile country and network codes already defined by ITU-T.

```
~/openairinterface5g/openair3/UTILS/mcc_mnc_itu.c

const mcc_mnc_list_t mcc_mnc_list[] = {
  {268, "08"},
...
};
```

- OAI's eNB is configured using a file with a certain set of parameters, which is used when running the eNB's lte-softmodem. A different file is required for each type of LTE duplex mode (FDD or TDD), as well as for different E-UTRA bands, downlink and uplink frequencies, number of RBs, power parameters, RF front-end used, network identification (PLMN-ID). The eNB's computer and the EPC's computer communicate via an Ethernet cable. Their IPs are 192.168.1.17 and 192.168.1.16, respectively.

  As several frequency bands and duplex modes were tested in this dissertation (FDD Band 1, 5, 7 and 31, and TDD Band 38 and 40), a total of six files were required. Only the required modifications will be presented for FDD Band 1 running on an USRP, `enb.band1.tm1.usrpb210.conf`, not the entire configuration file.

```
$OPENAIR_DIR/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band1.tm1.usrpb210.conf

////////// Identification parameters:
...
tracking_area_code = "1";
mobile_country_code = "268";
mobile_network_code = "08";

////////// Physical parameters:
component_carriers = (
 {
  frame_type           = "FDD";
  tdd_config           = 3;
  tdd_config_s            = 0;
  ...
  eutra_band              = 1;
  downlink_frequency      = 2140000000L;
  uplink_frequency_offset  = -190000000;
  ...
  N_RB_DL                 = 25;
  ...
  nb_antenna_ports        = 1;
  nb_antennas_tx          = 1;
  nb_antennas_rx          = 1;
  tx_gain                  = 90;
  rx_gain                  = 125;
  ...
  pdsch_referenceSignalPower = -24;
  pusch_p0_Nominal         = -90;
  pucch_p0_Nominal         = -96;
  ...
 }
);

srb1_parameters : {...};
SCTP : {...};

////////// MME parameters:
mme_ip_address = ({ipv4    = "192.168.1.16";
                   ipv6      = "192:168:30::17";
                   active    = "yes";
                   preference = "ipv4";
                  });

NETWORK_INTERFACES : {
   ENB_INTERFACE_NAME_FOR_S1_MME        = "eth1";
   ENB_IPV4_ADDRESS_FOR_S1_MME          = "192.168.1.17/24";
   ENB_INTERFACE_NAME_FOR_S1U           = "eth1";
   ENB_IPV4_ADDRESS_FOR_S1U             = "192.168.1.17/24";
   ENB_PORT_FOR_S1U                     = 2152;
};

log_config : {...};
```

C.3 EPC's computer

- Check if the 4.7.7-oaiepc kernel is installed. `uname -a` should output:

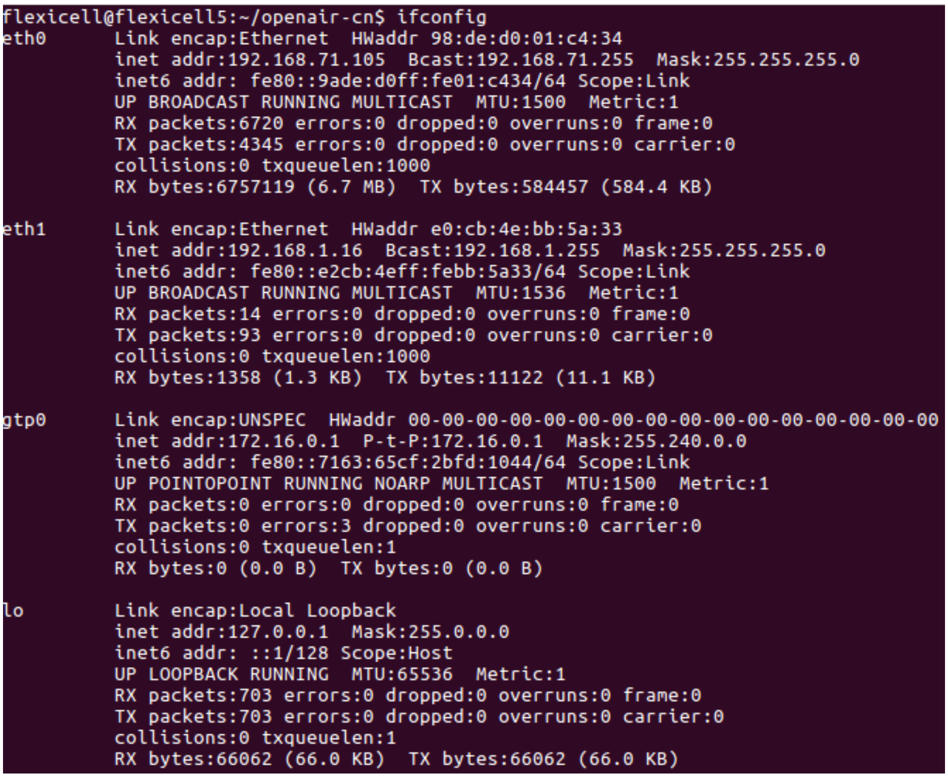
```
uname -a

   Linux flexicell5 4.7.7-oaiepc #1 SMP Sun Oct 16 22:45:31 CST 2016 x86_64
       x86_64 x86_64 GNU/Linux
```

- Confirm the develop branch is the one in use (indicated by the asterisk).

```
cd openair-cn && git branch -vvv

   * develop 724542d [origin/develop] Merge branch
       'feature-53-support-enterprise-linux-7-develop' into develop
     master  c5cad9d [origin/master] Fix test compilation
```

- Configure the network interfaces. In this case eth1 is attributed a manual IP address (192.168.1.16), and eth0 an automatic (DHCP) address, as shown in Figure C.3:
- Using automated build scripts, in openair-cn's `scripts` folder, the OAI EPC components (SPGW, HSS, MME) are installed, along with additional required packages (MySQL-server, phpmyadmin, freeDiameter, etc.), using the -i flag.

```
cd openair-cn/scripts/
./build_mme -i
./build_hss -i
./build_spgw -i
```



```
flexicell@flexicell5:~/openair-cn$ ifconfig
eth0      Link encap:Ethernet  HWaddr 98:de:d0:01:c4:34
          inet addr:192.168.71.105  Bcast:192.168.71.255  Mask:255.255.255.0
          inet6 addr: fe80::9ade:d0ff:fe01:c434/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6720 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4345 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6757119 (6.7 MB)  TX bytes:584457 (584.4 KB)

eth1      Link encap:Ethernet  HWaddr e0:cb:4e:bb:5a:33
          inet addr:192.168.1.16  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::e2cb:4eff:febb:5a33/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1536  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:93 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1358 (1.3 KB)  TX bytes:11122 (11.1 KB)

gtp0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.16.0.1  P-t-P:172.16.0.1  Mask:255.240.0.0
          inet6 addr: fe80::7163:65cf:2bfd:1044/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:3 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:703 errors:0 dropped:0 overruns:0 frame:0
          TX packets:703 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:66062 (66.0 KB)  TX bytes:66062 (66.0 KB)
```

Figure C.3: EPC's computer `ifconfig` Linux command.

- After installation, copy the EPC configuration files to /usr/local/etc/oai/:

```
sudo mkdir -p /usr/local/etc/oai/freeDiameter
sudo cp ~/openair-cn/ETC/mme.conf /usr/local/etc/oai
sudo cp ~/openair-cn/ETC/hss.conf /usr/local/etc/oai
sudo cp ~/openair-cn/ETC/spgw.conf /usr/local/etc/oai
sudo cp ~/openair-cn/ETC/acl.conf /usr/local/etc/oai/freeDiameter
sudo cp ~/openair-cn/ETC/mme_fd.conf /usr/local/etc/oai/freeDiameter
sudo cp ~/openair-cn/ETC/hss_fd.conf /usr/local/etc/oai/freeDiameter
```

- Each configuration file is now ready for modification. Starting with the MME configuration file, where the GUMMEI, TAI, NAS integrity and ciphering algorithms, network interfaces and IP address used to communicate with the serving gateway are defined. Only the required modifications are displayed.

```
/usr/local/etc/oai/mme.conf

MME :
{
   REALM            = "openair5G.eur";
   PID_DIRECTORY    = "/var/run";
   ...
   INTERTASK_INTERFACE : {...};

   S6A : {
      S6A_CONF = "/usr/local/etc/oai/freeDiameter/mme_fd.conf"; # MME
          freeDiameter configuration file path
      HSS_HOSTNAME = "hss";
   };

   # ------- SCTP definitions - Number of streams to use in input/output
   SCTP : {
      SCTP_INSTREAMS = 2;
      SCTP_OUTSTREAMS = 2;
   };

   S1AP : {...};

   # ------- MME served GUMMEI
   GUMMEI_LIST = (
      {MCC="268" ; MNC="08"; MME_GID="4" ; MME_CODE="1";}
   );

   # ------- MME served TAI
   TAI_LIST = (
      {MCC="268" ; MNC="08"; TAC = "1";}
   );

   NAS :
   {
      # 3GPP TS 33.401 section 7.2.4.3 Procedures for NAS algorithm selection
      # Decreasing preference goes from left to right
      ORDERED_SUPPORTED_INTEGRITY_ALGORITHM_LIST = ["EIA2","EIA1","EIA0"];
      ORDERED_SUPPORTED_CIPHERING_ALGORITHM_LIST = ["EEA0","EEA1","EEA2"];

      # EMM TIMERS
      ...
   };

   NETWORK_INTERFACES :
   {
      # MME binded interface for S1-C or S1-MME communication (S1AP)
      MME_INTERFACE_NAME_FOR_S1_MME      = "eth1";
      MME_IPV4_ADDRESS_FOR_S1_MME        = "192.168.1.16/24";

      # MME binded interface for S11 communication (GTPV2-C)
      MME_INTERFACE_NAME_FOR_S11_MME     = "lo";
      MME_IPV4_ADDRESS_FOR_S11_MME       = "127.0.11.1/8";
      MME_PORT_FOR_S11_MME               = 2123;
   };

   LOGGING : {...}
};

S-GW : {
  SGW_IPV4_ADDRESS_FOR_S11 = "127.0.11.2/8"; # S-GW binded interface for S11
      communication (GTPV2-C)
};
```

- Moving on to SPGW's configuration file, where the network interfaces are configured, including the IP address pool for the UEs and the DNS IP addresses.

```
/usr/local/etc/oai/spgw.conf

S-GW : {
  NETWORK_INTERFACES :
  {
    # S-GW binded interface for S11 communication (GTPV2-C)
    SGW_INTERFACE_NAME_FOR_S11 = "lo";
    SGW_IPV4_ADDRESS_FOR_S11 = "127.0.11.2/8";

    # S-GW binded interface for S1-U communication (GTPV1-U)
    SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP = "eth1";
    SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP  = "192.168.1.16/24";
    SGW_IPV4_PORT_FOR_S1U_S12_S4_UP     = 2152;
  };

  INTERTASK_INTERFACE : {...};

  LOGGING : {...};
};

P-GW =
{
  NETWORK_INTERFACES :
  {
    # P-GW binded interface for SGi (Internet traffic)
    PGW_INTERFACE_NAME_FOR_SGI = "eth0";
    PGW_MASQUERADE_SGI      = "yes"; # the network will perform SNAT
  };
  IP_ADDRESS_POOL :
  {
    # Pool of IP addresses to be assigned to the UEs
    # First IPv4 address is reserved for the GTP network device on SPGW
    IPV4_LIST = ( "172.16.0.0/12" );
  };

  # DNS address communicated to UEs
  DEFAULT_DNS_IPV4_ADDRESS    =  "193.136.92.73";
  DEFAULT_DNS_SEC_IPV4_ADDRESS = "193.136.92.74";
};
```

- In HSS's configuration file, the MySQL configurations defined during the installation process are inserted, such as the username, password and database name.

```
/usr/local/etc/oai/hss.conf
```

```
HSS :
{
   # MySQL mandatory options
   MYSQL_server = "127.0.0.1"; # HSS S6a bind address
   MYSQL_user  = "root";     # Database server login
   MYSQL_pass  = "*******";     # Database server password
   MYSQL_db    = "oai_db";    # The database's name

   OPERATOR_key = "" ; # OP key, left blank as there are several options in
       the database

   # HSS's freeDiameter configuration file location
   FD_conf = "/usr/local/etc/oai/freeDiameter/hss_fd.conf";
};
```

- The HSS freeDiameter configuration file also needs modification, to specify a valid FQDN for the Diameter protocol, defined earlier.

```
/usr/local/etc/oai/freeDiameter/hss_fd.conf

Identity = "hss.openair5G.eur";
Realm = "openair5G.eur";
```

- The MME freeDiameter configuration file also needs to be provided with a valid FQDN for the Diameter protocol, along with a peer whom it should maintain a connection with, hss.

```
/usr/local/etc/oai/freeDiameter/mme_fd.conf

Identity  = "flexicell5.openair5G.eur";
Realm     = "openair5G.eur";

ConnectPeer = "hss.openair5G.eur" {ConnectTo = "127.0.0.1"; No_SCTP ; No_IPv6;
    Prefer_TCP; No_TLS; port = 3868; realm = "openair5G.eur";};
```

- Install HSS and MME S6a certificates for valid FQDNs defined in the freeDiameter configuration files.

```
cd ~/openair-cn/scripts
./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.openair5G.eur
./check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter/
    flexicell5.openair5G.eur
```

- Finally, the network components are ready to be compiled and built. Starting with the HSS, the database containing the subscriber's information has to be installed. It could be created by the user or adapted from a sample database file provided by OAI, `oai_db.sql`. It should contain three main tables, `mmeidentity`, `users`, and `pdn`, which will be briefly described later. After the HSS is built and the database is installed, it is the MME and SPGW's turn. Flag -c is used to clean previous build entries.

```
cd openair-cn/scripts
./build_hss -c
./run_hss -i  openair-cn/SRC/OAI_HSS/db/oai_db.sql
./build_mme -c
./build_spgw -c
```

The EPC is now ready for execution. Next, the UE's computer is configured.

C.4 UE's computer - Mini PC Intel NUC

- Check if the 4.8 low-latency kernel is installed, as in section C.2.
- Confirm the develop branch is the one in use, as in section C.2
- Using OAI's build wrapper, in openairinterface5g's cmake_targets folder, the UE is built, and additional required software packages are installed. The --UE flag compiles the UE's specific NAS parts.

```
cd openairinterface5g/
source oaienv
./cmake_targets/build_oai -I --eNB --UE -w USRP -c -C
```

- The new network's MCC (268) and MNC (08) must be added to the list of mobile country and network codes already defined by ITU-T, as in section C.2.
- A file containing a list of the known PLMNs of network operators must be altered. Information about the most important network operators in Portugal was added, along with the newly created network.

```
~/openairinterface5g/openair3/NAS/TOOLS/network.h

// Global Constants - PLMN network operator record index
#define FLEX_UE 1
#define VODAF   2
#define OPTIM   3
#define TMN     4

#define SELECTED_PLMN FLEX_UE

#define FLEX_UE_PLMN {6,2,0x0f,8,8,0} // 26808
#define VODAF_PLMN  {6,2,0x0f,8,1,0} // 26801
#define OPTIM_PLMN  {6,2,0x0f,8,3,0} // 26803
#define TMN_PLMN    {6,2,0x0f,8,6,0} // 26806

...

// Global Variables - List of PLMN network operator records
```

```
network_record_t network_records[] = {
   {26808, FLEX_UE_PLMN , "FLEX_UE" , "FLEX_UE" , 0x0001, 0xfffd},
   {26801, VODAF_PLMN , "VODAF"   , "VODAF"   , 0x0001, 0xfffd},
   {26803, OPTIM_PLMN , "OPTIM"   , "OPTIM"   , 0x0001, 0xfffd},
   {26806, TMN_PLMN   , "TMN"     , "TMN"     , 0x0001, 0xfffd},
};
```

- A file containing the OAI UE's known PLMNs and configurations regarding the user and abstracted USIM card is created. The IMSI, IMEI, OPc and K values need to match the ones programmed on the database. The selected operating PLMN-ID is the new network, 26808.

```
~/openairinterface5g/openair3/NAS/TOOLS/ue_flexible_test.conf

# List of known PLMNS
PLMN: {
   PLMN0: {
      FULLNAME="FLEX_UE";
      SHORTNAME="FLEX_UE";
      MNC="08";
      MCC="268";
   };
   PLMN1: {
      FULLNAME="VODAF";
      SHORTNAME="VODAF";
      MNC="01";
      MCC="268";
   };
   PLMN2: {
      FULLNAME="OPTIM";
      SHORTNAME="OPTIM";
      MNC="03";
      MCC="268";
   };
   PLMN3: {
      FULLNAME="TMN";
      SHORTNAME="TMN";
      MNC="06";
      MCC="268";
   };
};

UE0:
{
   USER: {
      IMEI="356092040793012";
      MANUFACTURER="FLEXICELL";
      MODEL="LTE Android PC";
      PIN="0000";
   };

   SIM: {
        MSIN="0000000009";
        USIM_API_K="8baf473f2f8fd09487cccbd7097c6862";
        OPC="8e27b6af0e692e750f32667a3b14605d";
        MSISDN="351980000009";
    };

    # Home PLMN Selector with Access Technology
    HPLMN= "26808";

    # Operator PLMN List
    OPLMN_LIST = ("26808");

    # Operator controlled PLMN Selector with Access Technology
    OCPLMN_LIST = ("26801","26803","26806");
};
```

- As a new file regarding the UE configuration file was created, its path in the `build_oai` script needs to be added.

```
~/openairinterface5g/cmake_targets/build_oai

conf_nvram_path=$OPENAIR_DIR/openair3/NAS/TOOLS/ue_flexible_test.conf
```

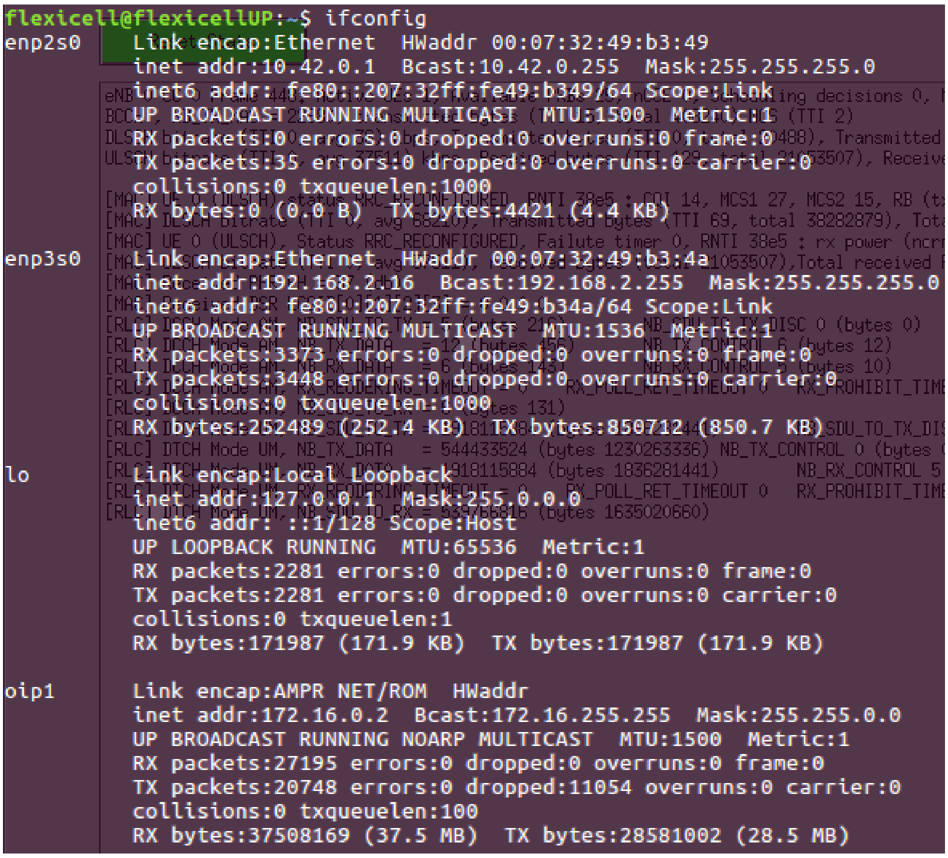
To finalize the UE's installation process, it must be re-built to save every modification.

```
./build_oai -w USRP --eNB --UE -c -C
```

C.5 UE's Single-Board Computer (SBC) - UP Squared

The installation process is different for the UP Squared. Due to its lower processing capabilities, the Turbo mode can't be disabled, for example. Instead, each one of the three UE threads will be mapped into an individual processor core, leaving one free for Linux's non real-time processing. OAI's deadline scheduler will be disabled and a FIFO scheduler is used instead, as the deadline scheduler generates errors with the USRP/UHD driver. The SBC will be controlled remotely by the eNB's computer, via a SSH session, to relieve the processor from graphical processing. Any other peripherals that are not the USRP B200mini will also be removed.

- Check if the 4.8 low-latency kernel is installed, as in section C.2.
- Confirm the develop branch is the one in use, as in section C.2.
- Configure the board's network interface, in this case `enp3s0`, with a manual IP address (192.168.2.16), as in Figure C.4, and an Ethernet cable will be connected to the eNB's computer (for SSH purposes). As shown in Figure C.2, eNB's eth0 was configured as 192.168.2.17.
- Using OAI's build wrapper, in openairinterface5g's cmake_targets folder, the UE is built without the deadline scheduler, and additional required software packages are



```
flexicell@flexicellUP:~$ ifconfig
enp2s0    Link encap:Ethernet  HWaddr 00:07:32:49:b3:49
          inet addr:10.42.0.1  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::207:32ff:fe49:b349/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4421 (4.4 KB)

enp3s0    Link encap:Ethernet  HWaddr 00:07:32:49:b3:4a
          inet addr:192.168.2.16  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::207:32ff:fe49:b34a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1536  Metric:1
          RX packets:3373 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3448 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:252489 (252.4 KB)  TX bytes:850712 (850.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2281 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2281 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:171987 (171.9 KB)  TX bytes:171987 (171.9 KB)

oip1      Link encap:AMPR NET/ROM  HWaddr
          inet addr:172.16.0.2  Bcast:172.16.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:27195 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20748 errors:0 dropped:11054 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:37508169 (37.5 MB)  TX bytes:28581002 (28.5 MB)
```

Figure C.4: UE's SBC `ifconfig` Linux command.

installed. The --disable-deadline flag disables the deadline scheduler, using a FIFO scheduler instead.

```
cd openairinterface5g/
source oaienv
./cmake_targets/build_oai -I --eNB --UE -w USRP -c -C --disable-deadline
```

- Add the new network's MCC (268) and MNC (08) to the list of mobile country and network codes already defined by ITU-T, as in section C.2.

- A file containing a list of the known PLMNs of network operators must be altered, as in section C.4. Information about the most important network operators in Portugal was added, along with the newly created network.

- A file containing the OAI UE's known PLMNs and configurations regarding the user and abstracted USIM card is created, as in section C.4. The IMSI, IMEI, OPc and K values need to match the ones programmed on the database. The selected operating PLMN-ID is the new network, 26808.

- As a new file regarding the UE configuration file is created, its path in the `build_oai` script must be added, as in section C.4. To finalize the UE's installation process, it has to be re-built to save every modification, without forgetting the --disable-deadline flag.

- For the core mapping, core 0 is used for non real-time processing, core 1/2 for processing even/odd subframes, and core 3 for the I/Q samples acquisition and internal scheduling. These core functionalities are already set in the function `~/targets/RT/USER/lte-ue.c` in the current develop branch, and when the system is executed, the configuration is sent to the kernel.

  To start the system and set the three cores for their defined functionalities, the cset shield command is used. First, a shield is set on the last three cores, meaning all other non real-time tasks are moved to core 0. Then, the governor is changed to `performance` in all cores, to operate at maximum performance. To improve hardware latency, the USRP/USB IRQs can also be moved to core 3, which manages the I/Q samples.

```
sudo bash
cset shield --force --kthread on -c 1-3

for f in /sys/devices/system/cpu/cpu[0-9]* ; do
   echo "performance" > $f/cpufreq/scaling_governor
done

echo ’3’ > /procirq/30/smp_affinity_list
```

  The UE running on the SBC is now ready to run.

C.6 Commercial UE’s Configuration

For the Samsung Galaxy S4 and OnePlus 3, all is needed are blank programmable USIM cards. Using Gemalto’s card reader and the Card Admin tool, the USIM cards can be programmed to work with the new network. This way, all parameters are available to be user, unlike the regular operator’s USIM cards. The parameters used for both cards were previously presented, in Table 6.3.

Then, a new APN profile must be created, for LTE mobile data access, and it must match the one to be assigned to the UE in the HSS database. For Samsung Galaxy S4, a new APN profile is created by going to Settings-Mobile Network Settings-Access Point Names-Add a new APN. A name is assigned to the profile, eur. The APN itself is oai.ipv4, and must the one on the database. At the Bearer field, the option LTE is chosen, and the Mobile virtual network operator field is completed with 26808x. The APN profile is then saved and selected.

C.7 User Registration on HSS Database

MySQL and phpMyAdmin were installed during the EPC’s build process. phpMyAdmin is just a way to observe and modify the MySQL database on a browser. The programmed USIM cards and OAI UE abstracted USIM card parameters need to be added to the HSS database, in the `oai_db.users` table.

- In EPC’s computer, add the users Samsung Galaxy S4, OnePlus 3 and OAI UE parameters to the table `oai_db.users`. Only the Galaxy S4 and OAI UE parameters are shown here.

```
mysql -u root -p
use oai_db;
INSERT INTO users (‘imsi‘, ‘msisdn‘, ‘imei‘, ‘imei_sv‘, ‘ms_ps_status‘,
    ‘rau_tau_timer‘, ‘ue_ambr_ul‘, ‘ue_ambr_dl‘, ‘access_restriction‘,
    ‘mme_cap‘, ‘mmeidentity_idmmeidentity‘, ‘key‘, ‘RFSP-Index‘, ‘urrp_mme‘,
    ‘sqn‘, ‘rand‘, ‘OPc‘) VALUES (’268080000000002’, ’351980000002’, NULL,
    NULL, ’NOT_PURGED’, ’120’, ’50000000’, ’100000000’, ’47’, ’0000000000’,
    ’1’, 0x8BAF473F2F8FD09487CCCBD7097C6862, ’1’, ’0’, ’’,
    0x00000000000000000000000000000000, 0x8E27B6AF0E692E750F32667A3B14605D);

INSERT INTO users (‘imsi‘, ‘msisdn‘, ‘imei‘, ‘imei_sv‘, ‘ms_ps_status‘,
    ‘rau_tau_timer‘, ‘ue_ambr_ul‘, ‘ue_ambr_dl‘, ‘access_restriction‘,
    ‘mme_cap‘, ‘mmeidentity_idmmeidentity‘, ‘key‘, ‘RFSP-Index‘, ‘urrp_mme‘,
    ‘sqn‘, ‘rand‘, ‘OPc‘) VALUES (’268080000000009’, ’351980000069’,
    ’356092040793012’ , NULL, ’NOT_PURGED’, ’120’, ’50000000’, ’100000000’,
    ’47’, ’0000000000’, ’1’, 0x8BAF473F2F8FD09487CCCBD7097C6862, ’1’, ’0’, ’’,
    0x00000000000000000000000000000000, 0x8E27B6AF0E692E750F32667A3B14605D);
```

- Update the `oai_db.mmeidentity` table with MME’s hostname, and the `oai_db.pdn` table with the PDN configurations, including the APN. Only the Galaxy S4 and OAI UE configurations are shown here, for the `pdn` table.

```
INSERT INTO mmeidentity
    (‘idmmeidentity‘,‘mmehost‘,‘mmerealm‘,‘UE-reachability‘) VALUES
    (’1’,’flexicell5.openair5G.eur’,’openair5G.eur’,’0’);

INSERT INTO pdn (‘id‘, ‘apn‘, ‘pdn_type‘, ‘pdn_ipv4‘, ‘pdn_ipv6‘,
    ‘aggregate_ambr_ul‘, ‘aggregate_ambr_dl‘, ‘pgw_id‘, ‘users_imsi‘, ‘qci‘,
    ‘priority_level‘,‘pre_emp_cap‘,‘pre_emp_vul‘, ‘LIPA-Permissions‘) VALUES
    (’1’, ’oai.ipv4’, ’IPV4’, ’0.0.0.0’, ’0:0:0:0:0:0:0:0’, ’50000000’,
    ’100000000’, ’3’, ’268080000000002’, ’9’, ’15’, ’DISABLED’, ’ENABLED’,
    ’LIPA-ONLY’);

INSERT INTO pdn (‘id‘, ‘apn‘, ‘pdn_type‘, ‘pdn_ipv4‘, ‘pdn_ipv6‘,
    ‘aggregate_ambr_ul‘, ‘aggregate_ambr_dl‘, ‘pgw_id‘, ‘users_imsi‘, ‘qci‘,
    ‘priority_level‘,‘pre_emp_cap‘,‘pre_emp_vul‘, ‘LIPA-Permissions‘) VALUES
    (’1’, ’oai.ipv4’, ’IPV4’, ’0.0.0.0’, ’0:0:0:0:0:0:0:0’, ’50000000’,
    ’100000000’, ’3’, ’268080000000009’, ’9’, ’15’, ’DISABLED’, ’ENABLED’,
    ’LIPA-ONLY’);
```

C.8 Custom/Unlicensed Frequency Bands

In case a custom/unlicensed frequency band is going to be used for transmission, additional steps are required in the eNB and UE implementations. Two files have to be altered on the UE side, `~/targets/RT/USER/lte-ue.c` and `~/openair2/ENB-APP/enb_config.c`. This last file also needs to be changed on the eNB side. The modifications are exactly the same for each file, and consist on adding the custom E-UTRA band, along with its uplink and downlink operating bands. In this case, FDD band 31 was added (UL 452.5-457.5 MHz, DL 462.5-467.5 MHz).

```
...
static const eutra_band_t eutra_bands[] = {
   ...
   {31, 452500 * KHz, 457500 * KHz, 462500 * KHz, 467500 * KHz, FDD},
   ...
};
...
```

In case of wanting to transmit on an unlicensed band, it is necessary to modify some extra parameters, besides the `eutra_bands` described earlier and the `eutra_band[]`, `downlink_frequency` and `uplink_frequency_offset` in the configuration file. As the unlicensed spectrum is for frequencies higher than usual (e.g. band 46, 5150-5925 MHz), the data types used for several functions must be changed from `uint32_t` and `int32_t` to `uint64_t` and `int64_t`. These functions are `~/targets/RT/USER/lte-softmodem.c` (parameters `downlink_frequency` and `uplink_frequency_offset`), `~/targets/RT/USER/lte-ue.c` (parameters `ul_min`, `ul_max`, `dl_min` and `dl_max`) and `~/targets/RT/USER/enb_config.c` (all the previous changes on data types).

C.9 Testbed Execution

Every computer/SBC is now correctly configured and prepared to run OAI. Several testbeds can be implemented, but the ones being described in this guide are solely the implemented test setups used in this dissertation (in section **??**). The radio configurations were already discussed in Chapter 5 of this dissertation.

The first and most stable setup to be implemented is the OAI eNB and OAI EPC, connected with a commercial UE. The second setup switches the commercial UE with the OAI UE, running in a powerful mini computer. The third setup trades the UE's computer with a more compact and less powerful single-board computer.

In order to obtain a functional network, the system has to be executed in a specific order: core network → access network → user equipment. For the core network, HSS must always be executed first then MME. S/P-GW can be ran in the first or last place.

C.9.1 Commercial Off-The-Shelf (COTS) UE

- In EPC's computer, the running scripts are executed for the three components. The MME is now waiting for eNB's attach.

  ```
  cd openair-cn/scripts/
  sudo ./run_spgw
  sudo ./run_hss
  sudo ./run_mme
  ```

- eNB's computer is connected to an USRP B210 via USB 3.0. The eNB is built, and the `lte-softmodem` wrapper can now be executed.

  ```
  cd ~/openairinterface5g/
  source oaienv
  ./cmake_targets/build_oai -w USRP --eNB -x -c -C
  ```

```
sudo -E ./targets/bin/lte-softmodem.Rel14 -d -O
    ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band1.tm1.
    usrpb210.conf
```

Summary of the used command flags:

- -x: adds a software oscilloscope (softscope) to the produced binaries.
- -E: a sudo flag, used to preserve the existing environment variables.
- -d: enables the softscope and L1/L2 stats.
- -O: provides the path for the configuration file.

The eNB should now be attached to the EPC, as shown on the left side of Figure 6.5.

- For the commercial UE, Samsung Galaxy S4 or OnePlus 3, the most stable mode to establish a connection with the network is to keep the phone in airplane mode until the eNB attaches correctly to the EPC. The airplane can then be disabled, and the mobile data enabled. If every step was followed accordingly, the phone should be attached to the network (see right side of Figure 6.5) and able to browse the Internet, using 4G mobile data.

C.9.2 OAI UE

- In EPC's computer, the procedure is the same as in section C.9.1.
- In eNB's computer, connected to an USRP B210 via USB 3.0, the procedure is the same as in section C.9.1.
- UE's computer is connected to an USRP B200mini via USB 3.0. The UE softmodem is built and executed with several flags. Between the two, the `ue_ip.ko` kernel module has to be installed, to bring up the oip1 interface, which will be attributed an IP from the S/P-GW, through MME.

```
cd ~/openairinterface5g/
source oaienv
./cmake_targets/build_oai -w USRP --eNB --UE -x -c -C
cd targets/bin/
sudo insmod ue_ip.ko
sudo ./lte-softmodem.Rel14 -U -C2140000000 -r25 --ue-scan-carrier --ue-txgain
    90 --ue-rxgain 125 -d
```

Summary of the lte-softmodem flags:

- -U: configure the execution for UE mode.
- -C: choose the center frequency for transmission, in Hz.
- -r: select the number of resource blocks: 25, 50 or 100.
- --ue-scan-carrier: scans for the best carrier frequency.
- --ue-txgain and --ue-rx-gain: change the transmission and reception gain, in dB, determined via calibration.

- -d: enable the softscope and a stats window.
- -T: not used in this execution, but this represents TDD transmission and is added to the lte-softmodem.Rel14 line.

The UE should now be attached to the network, through the radio link established with the virtual network interface oip1, created during execution. This can be confirmed by checking if the oip1 has an assigned IP.

- The UE is attached to the network. However, the Internet can't be browsed. The DNS configurations are not passed on to the OAI UE as in commercial UEs; first, oip1 must be added as the computer's default route, followed by the network's DNS IP addresses.

```
sudo ip route flush cache
sudo ip route add default dev oip1 via 172.16.0.1
echo -e "nameserver 193.136.92.73\nnameserver 193.136.92.74" | sudo tee
    /etc/resolv.conf
sudo /etc/init.d/networking restart
```

Between executions, it is best to remove the oip1 kernel module (`sudo rmmod ue_ip`) and install it again (`sudo insmod ue_ip.ko`), to avoid faulty UE runs.

C.9.3 OAI UE on a Single-Board Computer (SBC)

- In EPC's computer, the procedure is the same as in section C.9.1.
- In eNB's computer, connected to an USRP B210 via USB 3.0, the procedure is the same as in section C.9.1.
- UE's single-board computer, the UP Squared, is connected to an USRP B200mini via USB 3.0. As in the previous section, the UE softmodem is built and executed with several flags, with the kernel module installed in between. The execution is a bit different, as the threads are running in individual CPU cores.

```
cd ~/openairinterface5g/
source oaienv
./cmake_targets/build_oai -w USRP --eNB --UE -c -C --disable-deadline
cd targets/bin/
sudo insmod ue_ip.ko
sudo cset shield ./lte-softmodem.Rel14 -- -U -C2140000000 -r25
    --ue-scan-carrier --ue-txgain 90 --ue-rxgain 125
```

`cset shield` allows the user to choose what to run on the shielded CPU cores. The distribution of the cores was already set during the installation process. This time, no software oscilloscope or stats windows are compiled and executed, as no graphical interface is used.

The UE is now attached to the network, through a radio link established with the virtual network interface oip1. This can be confirmed by checking if an IP address was assigned to it, as in Figure C.4.

- Same as in the previous section, the UE is not attributed with the network's DNS IP addresses, or a default route, so the Internet can't be browsed yet. They have to be manually added first, just like in section C.9.2.

The UE is now connected to the network and with LTE mobile data access.

C.9.4 Sharing the network

The main goal of this dissertation was to have a portable device that could provide Internet access to surrounding equipments. If at least one Ethernet port is free, or a Wi-Fi module is available, the LTE connection from the OAI UE can be shared.

C.9.4.1 Ethernet

Ubuntu's Network Manager GUI allows the user to share the existent Internet connection with other computers: Edit Connections → Add a new Ethernet wired connection → IPv4 Settings → Shared to other computers.

In case there's no access to the Network Manager GUI, e.g. PC controlled with a remote connection, the routing/SNAT has to be done manually, attributing an IP address for the shared network to a network interface and forwarding its traffic to oip1.

```
sudo ip addr add 10.42.0.1/24 dev eth0
sudo iptables -t nat -F POSTROUTING
sudo iptables -t nat -A POSTROUTING -o oip1 -s 10.42.0.0/24 ! -d 10.42.0.0/24 -j
    MASQUERADE
sudo sysctl -w net.ipv4.ip_forward=1
sudo /etc/init.d/networking restart
```

The Ethernet cable can now be connected to an AP or a single PC. If it's just one computer, a new wired connection must be added, and its IP configured manually to an address in the range of the shared network's IP (say, 10.42.0.2/24), with the gateway and DNS server being 10.42.0.1.

If it's to be connected to an AP, it must be configured in LAN mode and with the DHCP option turned off. If no password is configured, free Internet is now provided with surrounding devices.

C.9.4.2 Wi-Fi module

If using a computer with a Wi-Fi module, the Internet can be shared by simply creating a Wi-Fi hotspot that uses the LTE mobile data provided by the oip1 interface.

Appendix D

3D Printed Prototype Holder

In order to create a compact and portable solution for the flexible LTE UE proposed in this dissertation, a prototype holder for the UP Squared and USRP B200mini was designed and 3D printed.

The USRP B200mini is a small board, the size of a business card, but its performance is very similar to the regular sized USRP B200/B210 boards. Because of this, the power dissipation is much worse on the B200mini and the board temperature rises very quickly when the system is being run.

On the other hand, and because the UP Squared was set up with an active fan instead of the original passive heatsink, its power dissipation is very good. The CPU cores' internal temperatures never going above 50°C, even when the system is running. The air expelled by the fan is still quite cold, which means the USRP could benefit from it. This was taken into account when modelling the 3D holder, facing the UP Squared fan towards the USRP.

The finished 3D model, measuring 10.1 x 10.3 x 8.3 cm, is presented in Figure D.1. Tinkercad [Tin17], a free and online 3D CAD design tool, was used for the design.

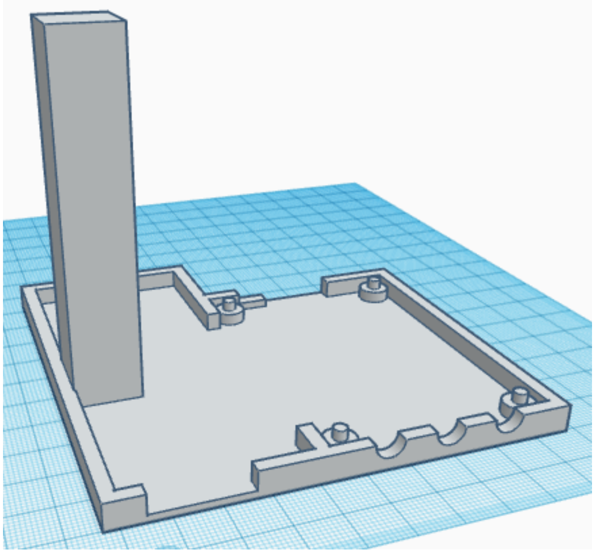


Figure D.1: Finished 3D model of the prototype holder.

In Figure D.2, the printed prototype is displayed. On the left, the 3D holder is pictured.

On the right, the prototype is holding the USRP B200mini and UP Squared boards. The USRP is fitted in the designed protuberances, and the UP Squared is screwed to the pillar.

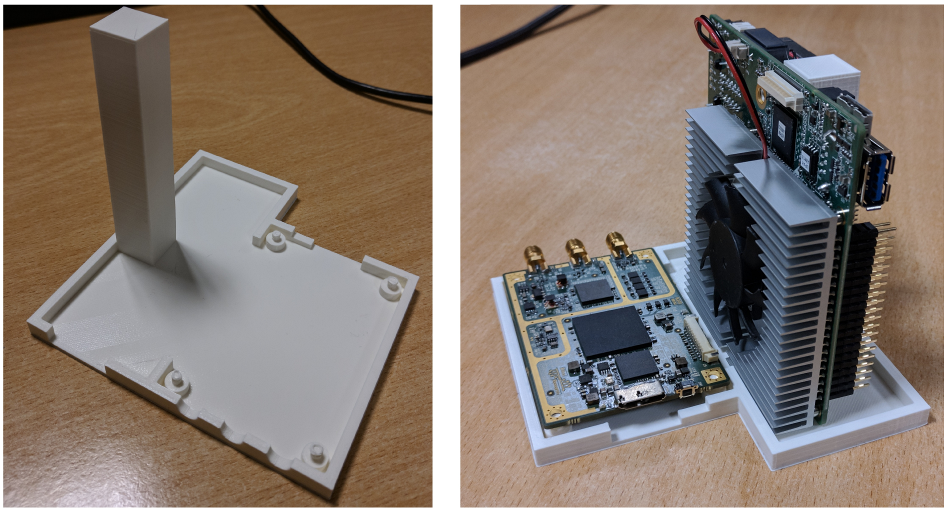


Figure D.2: Finished 3D prototype holder. On the left, without the boards. On the right, holding the UP Squared and USRP B200mini.