ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA

DIPARTIMENTO di
INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE
"Guglielmo Marconi"
DEI

CORSO DI LAUREA/LAUREA MAGISTRALE IN
TELECOMMUNICATIONS ENGINEERING

TESI DI LAUREA
in
NETWORK DESIGN M

EMULATION OF A PRIVATE NETWORK ENABLED BY SDN IN OPEN RAN

CANDIDATA

FREICY DANIELA VILLALVA CHOQUE

RELATORE

Chiar.ma Prof.ssa Carla Raffaelli

CORRELATORE

Ing. Giacomo Croce

Anno Accademico
2021/2022

Sessione
III

Contents

1 – Introduction

Although in the world the focus is on the Fifth Generation (5G) and other future generations, the industry needs to focus separately on the coverage of the Fourth Generation (4G) first. One of the reasons why the Long-Term Evolution mobile technology will continue to evolve is that, in general, the average user does not need 5G technology because its benefits are not substantial for consumer applications. The number of subscribers for the LTE technology will, therefore, continue to increase.

Open RAN is able to deploy both LTE and NR mobile technologies. The deployment of Open Radio Access Networks translates in costs reduction, as well as increasing flexibility, scalability and reliability. The O-RAN Alliance has promised to open a new ecosystem that is centralized on the software and where the hardware becomes a white box. This new software-centered ecosystem allows to obtain systems in which most, or all, of the RAN MAC and PHY layers are implemented via software.

Many mobile network operators (MNOs) throughout the world are, therefore, now testing and deploying Open RAN. Moreover, the approaches of Software-Defined Networking (SDN), such as virtualization and vendor neutrality, offer new and efficient ways to manage the mobile network which are extremely relevant for Open RAN. Open RAN, ergo, is part of the transformation in deploying SDN.

SDN brings flexibility to Open RANs, allowing MNOs to simply evolve their deployed private LTE network into a private NR network.

For all the reasons mentioned so far, the goal of this thesis is to emulate a Private LTE mobile Network. The emulated private network will be deployed in Open Radio Access Network and it will be enabled by Software-Defined Networking.

Blank page.

2 – Overview of LTE, Open RAN and SDN

2.1 - Long-Term Evolution

Mobile Network are increasingly based on an Internet Protocol network architecture. The success of the Global System for Mobile communications (GSM) was due to its foundation of circuit switching, being able to provide voice services over cellular networks. The first mobile Internet services had some limitations because of the processing capacity of terminals and a limited bandwidth on the radio interface. These limitations have been overcome with the evolution of the Radio Access Networks (RANs) which are able to provide high data rate thanks to the HSPA (High-Speed Packet Access) and Long-Term Evolution (LTE) technologies.

The core network is the part that links the Internet and mobile communications networks. The evolution of the core network, more commonly called EPC (Evolved Packet Core), is fundamental for the mobile broadband revolution. The Evolved Packet System (EPS) involves the radio access, the core network and the terminals that encompass all the mobile system.

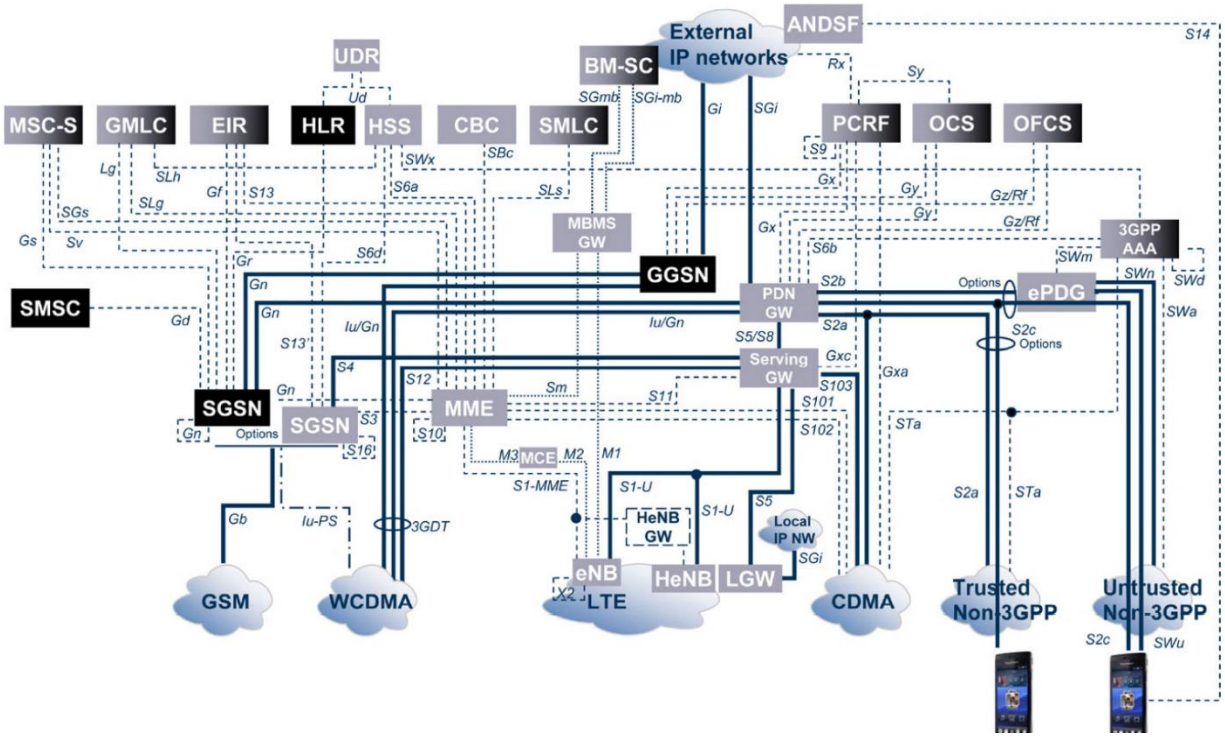


Figure 2.1.1: Architecture of the Evolved Packet Network.

The core network is divided in multiple domains: circuit core, packet core and IP Multimedia Subsystem, these domains consist of nodes and functions that interwork with each other over a certain number of interfaces. The circuit core domain provides support for circuit-switched services over GSM and WCDMA. The packet core domain provides support for packet-switched services over GSM, WCDMA, HSPA, LTE and non-3GPP access networks. The IMS domain provides support for multimedia sessions based on the SIP protocol (Session Initiation Protocol). The IMS domain uses the IP connectivity furnished by the functions in the packet core domain. The core network also includes the subscriber data management domain where it is possible to find the data related to the subscribers using the services of the domain. Formally, the last domain, in the 3GPP specifications, is not a separate domain. The following figure illustrates the complete logical architecture developed for Evolved Packet Systems, this means that it is unlikely that a single network operator would utilize all these logical nodes and interfaces. This image does not show the IP infrastructure supporting the logical nodes.

At the core of the EPC architecture there are the functions required to support basic IP connectivity over LTE access. The basic EPC Architecture for LTE encompasses the HSS, the MME, the PDN gateway, the Serving gateway and the eNB. In an LTE radio network, there exists at least one eNodeB, namely the LTE base station, but in case of a large network scenario there might be several eNodeBs and they usually are connected through the X2 interface.

The eNB includes all features needed to realize wireless connection between user devices and the network. All eNodeBs are connected to at least one MME, which stands for Mobility Management Entity, over the S1-MME interface. The MME takes care of all LTE-related control plane signaling, including of course the mobility and security functions for devices and terminals attaching via the LTE radio access network. The MME is connected to the HSS, Home Subscriber Server, over the S6a interface because it relies on the subscription of the users in order to get IP connectivity. The HSS manages user data for user accessing the LTE radio access network.

The user data payload, i.e., the IP packets flowing to and from the end devices, are managed by the Serving gateway (SGW) and the PDN gateway (PGW), where PDN

stands for Packet Data Network. These two logical nodes are connected over the S5 or S8 interface.

The SGW is connected to the base stations, the eNBs, via the S1-U interface, while the PGW is the point of interconnection with the external IP networks via the SGi interface. The PDN gateway also supports the Quality of Service (QoS) for end-user IP services, for instance, the PGW handles the packet bearer operations.

The S5/S8 interfaces use the GPRS Tunneling Protocol (GTP). The GTP protocol is the de-facto protocol used to interconnect mobile networks and it is defined by the 3GPP standards to carry General Packet Radio Service (GPRS) with LTE networks.

There are some other important logical nodes such as: Policy and Charging Rules Function (PCRF), Online Charging System (OCS) and Offline Charging System (OFCS). The first logical node mentioned is in charge of policy and charging control, while the other two logical nodes support features related to charging of end-users.

It is also observable that MMEs are connected via the S10 interface. This interface is used when a device moves between two pools.

For LTE deployment, interworking with access networks that support IP connectivity is crucial. 3GPP has defined two options for how to interconnect LTE and WCDMA/HSPA or LTE and GSM/GPRS: Interworking based on Gn-SGSN and Interworking based on S4-SGSN.

For what regards the main solutions for voice, they are two: one solution is to use IMS mechanisms and the other solution is the circuit-switched. The first choice in used when calls happen without leaving the LTE network, namely VoLTE calls, while the second option is called CS Fallback and it is performed through users that leave temporarily the LTE network for the duration of the voice calls over GSM or WCDMA. In this thesis, VoLTE calls will be tested.

In the EPC architecture there are nodes and interfaces defined to support broadcasting of content to multiple users at the same time, this is based on a technology called eMBMS (enhanced Multimedia Broadcast and Multicast Service) and it is defined for WCDMA and LTE.

The network architecture includes support for determining the geographical position of an end-user device. In this regard the important logical nodes to look at are: Gateway Mobile Location Center (GMLC) and Serving Mobile Location Center (SMLC).

Other relevant logical nodes are: EIR (Equipment Identity Register) and CBC (Cell Broadcaster Center). The last node forwards to the MME a warning that needs to be broadcasted, while EIR is used by MME when a device attaches.

The work on LTE started in late 2004 by defining a set of targets. A key technology for LTE is OFDM (Orthogonal Frequency Division Multiplexing) transmission scheme, that it is used in downlink transmission, namely from eNodeBs to the end devices. This scheme allows to utilize the entire spectrum subdivided in a number of channels, where each channel carries one subcarrier. The advantage of this scheme is the robustness to multipath fading.

In order to meet these requirements, 3GPP added some requirements in the LTE Release 10. The following new and enhanced features were added:

- Carrier Aggregation, it allows combination of multiple carriers for communication with a single device. It increases the data rate and, since the used carriers do not need to be contiguous in frequency, also operators with fragmentated spectrum can achieve higher data rates. Three possible scenarios are possible for carrier aggregation: Inter CA, Intra CA with contiguous frequencies, Intra CA with non-contiguous frequencies.
- Enhanced multi-antenna support, with Release 10, more than one antenna ports can be set in order to allow parallel transmission layers. In downlink direction, up to eight antenna ports can be used and in uplink direction, up to four antenna ports can be used.
- Improved heterogeneous network support, it improves the management of inter-cell interference.
- LTE relaying, it introduces the concept of Donor and Relay Base Stations. Donor Base Stations serve both mobile terminals and Relay Base Stations, while Relay Base Stations serve only mobile terminals and they connect to the Donor Base Station for the backhauling link.

LTE has been designed to support a wide range of different frequency bands, but there is a limit to how many different frequency bands a single LTE device can support. In the 3GPP Technical Specification 36.101 version 10.1.0 there is a table that shows the different frequency bands for which LTE operation is specified.

E-UTRA Operating Band	Uplink (UL) Operating Band BS Receive UE Transmit F_{UL_low}–F_{UL_high}	Downlink (DL) Operating Band BS Transmit UE Receive F_{DL_low}–F_{DL_high}	Duplex Mode
1	1920–1980 MHz	2110–2170 MHz	FDD
2	1850–1910 MHz	1930–1990 MHz	FDD
3	1710–1785 MHz	1805–1880 MHz	FDD
4	1710–1755 MHz	2110–2155 MHz	FDD
5	824–849 MHz	869–894 MHz	FDD
6[1]	830–840 MHz	875–885 MHz	FDD
7	2500–2570 MHz	2620–2690 MHz	FDD
8	880–915 MHz	925–960 MHz	FDD
9	1749.9–1784.9 MHz	1844.9–1879.9 MHz	FDD
10	1710–1770 MHz	2110–2170 MHz	FDD
11	1427.9–1447.9 MHz	1475.9–1495.9 MHz	FDD
12	699–716 MHz	729–746 MHz	FDD
13	777–787 MHz	746–756 MHz	FDD
14	788–798 MHz	758–768 MHz	FDD
17	704–716 MHz	734–746 MHz	FDD
18	815–830 MHz	860–875 MHz	FDD
19	830–845 MHz	875–890 MHz	FDD
20	832–862 MHz	791–821 MHz	FDD
21	1447.9–1462.9 MHz	1495.9–1510.9 MHz	FDD
22	3410–3490 MHz	3510–3590 MHz	FDD
23	2000–2020 MHz	2180–2200 MHz	FDD
24	1626.5–1660.5 MHz	1525–1559 MHz	FDD
25	1850–1915 MHz	1930–1995 MHz	FDD
33	1900–1920 MHz	1900–1920 MHz	TDD
34	2010–2025 MHz	2010–2025 MHz	TDD
35	1850–1910 MHz	1850–1910 MHz	TDD
36	1930–1990 MHz	1930–1990 MHz	TDD
37	1910–1930 MHz	1910–1930 MHz	TDD
38	2570–2620 MHz	2570–2620 MHz	TDD
39	1880–1920 MHz	1880–1920 MHz	TDD
40	2300–2400 MHz	2300–2400 MHz	TDD
41	2496–2690 MHz	2496–2690 MHz	TDD
42	3400–3600 MHz	3400–3600 MHz	TDD
43	3600–3800 MHz	3600–3800 MHz	TDD

Figure 2.1.2: EUTRA Operating Bands.

Devices with LTE capabilities are classified in different categories depending on their capabilities in terms of:

- Peak rates.

- Modulation schemes.
- MIMO variants supported.

The following table shows the different categories and their maximum peak rates supported.

	3GPP Release							
	Rel. 8/9/10/11				*Rel. 10/11 Only*			
Category	1	2	3	4	5	6	7	8
Downlink peak rate (Mbit/s)	10	50	100	150	300	300	300	3000
Uplink peak rate (Mbit/s)	5	25	50	50	75	50	150	1500

Figure2.1.3: Categories of LTE devices.

In the EPS, there are different ways in order to obtain data and voice services. For what concerns data services, there are two ways of realizing messaging support with EPC: either using an IP-based solution or using the circuit-switched infrastructure. Examples of IP-based solution are the IMS-based messaging and SMS-over-IP. For what regards voice services, there are two different ways in which they can be realized for LTE users: one way is to use Circuit-Switched Fallback and another way is using IMS-bases VoLTE technologies.

The operator may provide access to different Packet Data Networks with different services. For example, a PDN could be a specific IP network set up by the telecom operator in order to provide specific services. It is possible to furnish multiple services in a single PDN, in this case, the device can access to a single PDN per time or it can have multiple PDN connections opened simultaneously. The end-device provides information about the Packet Data Network that the user wants to access during the attachment procedure. This information is carried in the Access Point Name (APN). If the device does not provide any APN during the attachment procedure, the network will use a default APN.

A Packet Data Network connection has at least one EPS bearer but it may also have multiple bearers for the sake of a good Quality of Service (QoS). The first bearer created when a PDN connection is established is the “Default Bearer”, this bearer lasts for the whole lifetime of the PDN connection and it usually has a default type of QoS. Additional EPS bearers, called Dedicated Bearers”, might be activated for a PDN connection.

Subscriptions are identified with an International Mobile Subscriber Identity (IMSI), each subscription is given a unique IMSI with maximum length of fifteen digits. The IMSI is composed of a Mobile Country Code (MCC), a Mobile Network Code (MNC) and a Mobile Subscriber Identity (MSIN). The MCC identifies the country, while the MNC identifies the network within the country. The IMSI is the permanent subscription identifier used in the HSS and it is stored in the USIM. There are also temporary subscriber identifiers called Globally Unique Temporary ID (GUTI) and they are used for many purposes.

The registration areas are labeled Tracking Areas (TAs), this concept allows a User Equipment to belong to different TAs of a list so that when the UE changes position within the list of TAs, it does not need to perform a TAU (Tracking Area Update).

It is important that EPS provides an efficient Quality of Service (QoS) solution given that mobile broadband operators want to provide multiple services, that will share radio and core network resources, over their packet-switched access networks.

Depending on the bit rate requirements, there can be different QoS requirements. Each EPS bearer has two QoS parameters associated with it: the QoS Class Identifier (QCI) and the Allocation and Retention Priority (ARP). The first one determines what user-plane (UP) treatment the IP packets should receive, while the second one indicates the control-plane (CP) treatment a bearer should receive.

Also, some Evolved Packet System bearers have associated bit rate parameters to support the allocation of a Guaranteed Bit Rate (GBR) when establishing the bearer.

In EPS, the Domain Name System (DNS) is used to store information both on mapping between the Access Point Name and the PDN gateway and on mapping between Tracking Area Identity and the Serving gateway. The Home Network Domain is derived from the IMSI and the EPC node DNS subdomain (the DNS zone) is derived by adding the word “node” to the beginning of the Home Network Domain.

IMS is defined as a subsystem within the mobile network architecture and at the core of this subsystem there is the Call Session Control Function (CSCF) which is the entity that handles the SIP signaling, namely it takes care of invoking applications and controlling the media path. The CSCF is divided in three entities: Proxy-CSCF, Serving-CSCF and Interrogating-CSCF.

An important node is the Session Border Controller (SBC) that is an IP gateway between the IMS domain and an external IP network. The SBC manages the IMS sessions and furnishes support for quality of the session and controlling security.

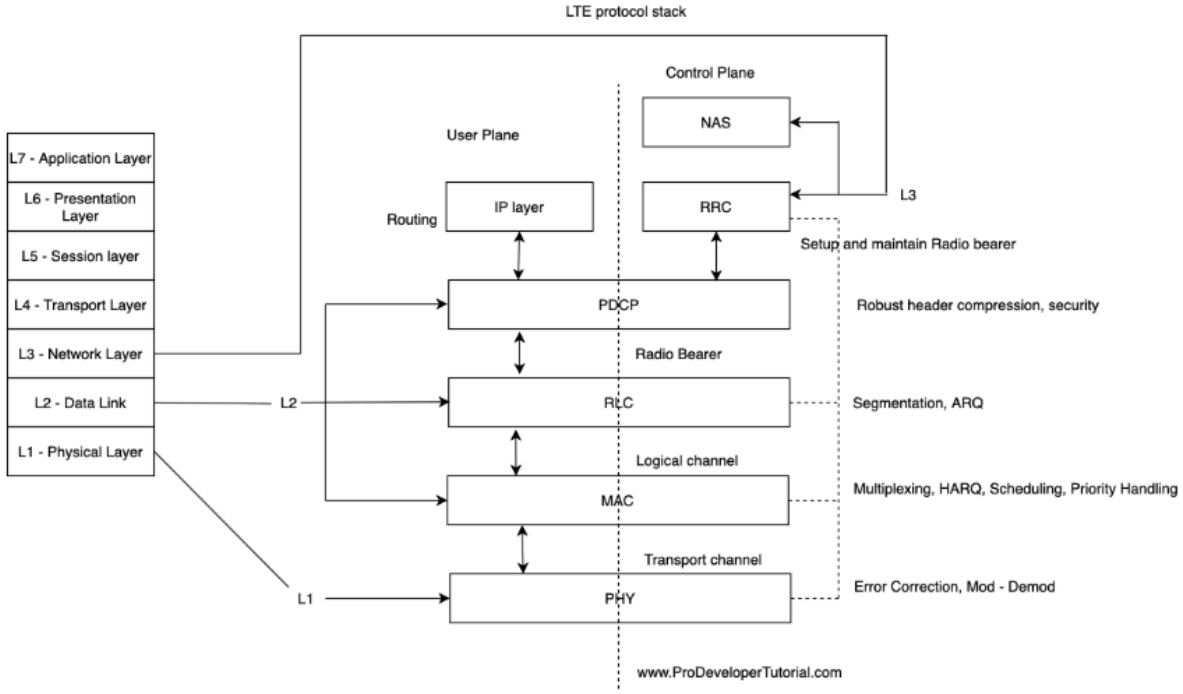


Figure 2.1.4: LTE protocol stack.

The protocol stack of LTE involves layers that have different functions, in the following lines they will be explained:

- Physical layer (PHY) carries the information from the MAC transport channels to the air interface. It takes care of the link adaptation, cell search and other measurements for the RRC layer.
- Medium Access Control Layer (MAC) is responsible for the mapping between logical channels and transport channels and for the multiplexing/demultiplexing of the MAC Service Data Units, error correction via HARQ and other tasks.
- Radio Link Control (RLC) is responsible for transfer of upper layer Packet Data Units, error correction via Automatic Retransmission request (ARQ), protocol error detection and recovery and other tasks.
- Radio Resource Control (RRC) is responsible for the broadcasting of System Information (SI) related to the NAS, paging, security management, establishment/maintenance/release of the point-to-point Radio Bearers and other tasks.

- Packet Data Convergence Protocol (PDCP) is responsible for the header compression/decompression, transfer of user and control plane data, access stratum ciphering/deciphering and others; also, PDCP is used for Signaling Radio Bearers (SRBs) and Data Radio Bearers (DRBs) mapped on the following logical channels: Dedicated Control Channel (DCCH) and Dedicated Traffic Channel (DTCH).
- Non-Access Stratum (NAS) protocols are the highest stratum of the Control Plane (CP) between the UE and the MME. They support the mobility of the user equipment and the session management procedures to start and maintain the IP connectivity between the UE and a PDN gateway, bearer management and other tasks.
- GTP User plane (GTP-U) protocol is used to carry signaling messages and encapsuled packet data units.

The User Plane protocol stack involves the following layers: PDCP, RLC, MAC and PHY. The Control Plane protocol stack comprises the following layers: NAS, RRC, RLC, MAC and PHY.

2..2 - Open RAN and SDN

Browsing the internet, it is possible to find a multitude of definitions for Open Radio Access Network, an example of definition is given by CISCO:

“An Open Radio Access Network is a nonproprietary version of the Radio Access Network (RAN) system that allows interoperation between cellular network equipment provided by different vendors.”

[https://www.cisco.com/c/en/us/solutions/what-is-open-ran.html]

In the traditional RAN system, hardware and software are proprietary. The “opening” of the RAN introduces the disaggregation of the Remote Radio Head (RRH) and the Baseband Unit (BBU); this disaggregation is possible thanks to the introduction of series of standards.

The RAN is disaggregated into units that can be containerized or virtualized thanks to the network virtualization and the interfaces among these units are open as well.

The implementation of Open RANs allows to the service providers a journey towards a RAN fully programmable, intelligent and multi-vendor. Summing up, Open RAN enables flexibility and interoperability.

Figure 2.2.1: Benefits of Open RAN.

Before “opening” the RAN, most of the network operators had to deploy networks using a single vendor in the interest of the performance of the network. Now, open RANs allow operators to buy from different vendors thanks to the use of open protocols and open interfaces.

Since the birth of mobile networks, it is the first time that a mobile operator can choose and custom its own software according to its own needs in order to have a network working at its best. The hardware, therefore, becomes only a commercial server where a specific software is running. The benefits are:

- Network agility.
- Independent testing of elements.
- Cost savings.
- Flexibility.
- Interoperability.
- Scalability.

The mobile network comprises the Radio Access Network and the Core Network. The RAN is the visible part and includes the antennas on the towers and the base stations. Thanks to the RAN the end device can connect to the core network through the digitalization of the received signal. The Core Network, first of all, authenticates the user. Only after being authenticated, the user can benefit from the services of the network. The core network, therefore, connects the final user to the Internet.

As mentioned previously, the RAN is now disaggregated into Radio Unit, Distributed Unit and Centralized Unit.

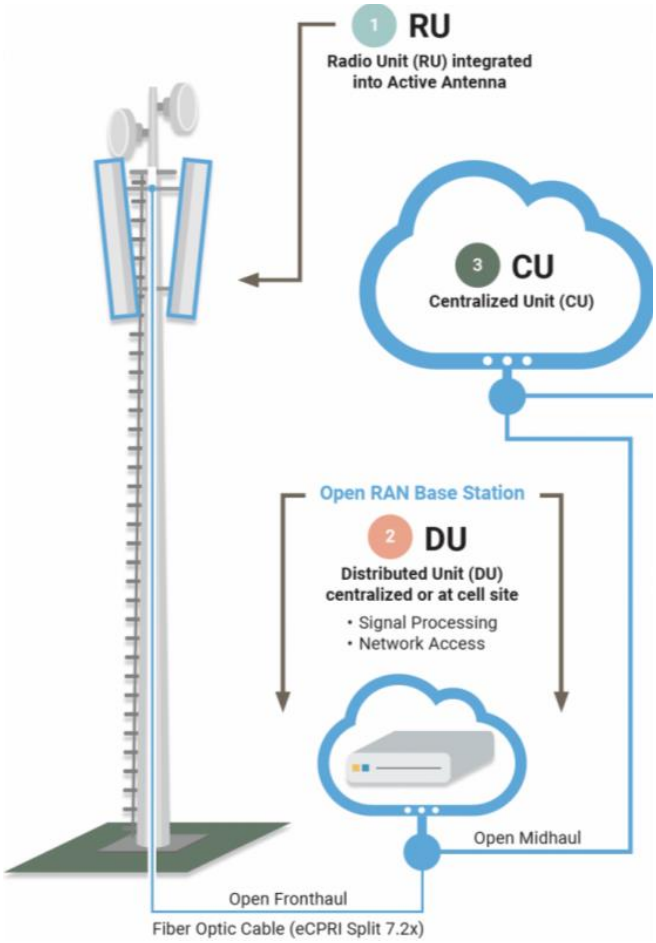


Figure 2.2.2: Components of disaggregated RAN.

The Radio Unit is usually located near or integrated into the antenna and it is where the signals are transmitted, received, amplified and digitized. The Distributed Unit consists of the real-time baseband processing functions. The Centralized Unit is where the packet processing functions are placed.

The Distributed and Centralized Units are the computation components of the base station and they are in charge of sending the digital signals from the RAN to the CN.

As it can be seen from the following picture, provided by Nokia, the DU can be positioned at the RU or concentrated in aggregated locations near the RU, while the CU is generally placed near to the core network.

The network virtualization together with the “opening” of the RAN allow to move some network functionalities from specific Hardwares to the cloud, which has its infrastructure relatively close to the RAN, in this way the latency is noticeably reduced.

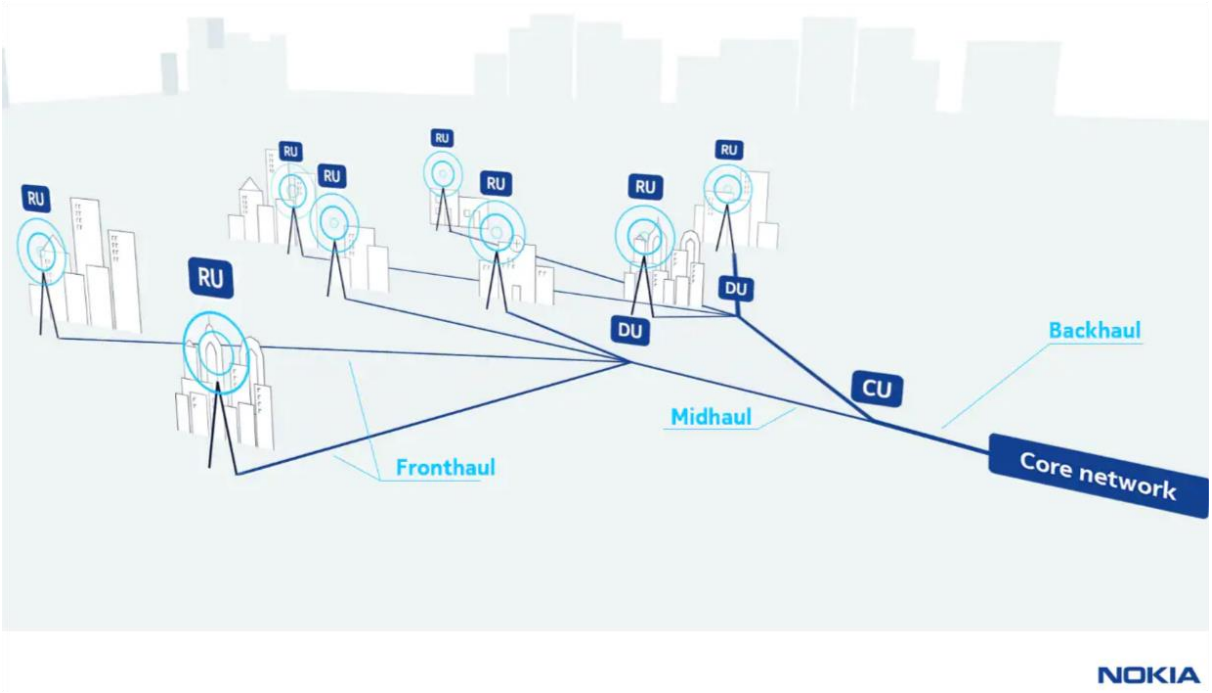


Figure 2.2.3: Moving network functionalities towards the cloud.

In the image representing the location of RU, DU and CU, it can be noticed the representation of the open interfaces defined by the ORAN Alliance, which are: the Fronthaul, Midhaul and Backhaul. The “opening” of these interfaces allows to virtualize the DU and the CU so that their virtualized software functions can be run on vendor-neutral hardware.

The Open RAN Alliance is a standardization organization for mobile networks that has fixed specific techniques for the sake of interoperability among different vendors. O-RAN is the de facto standard for open, disaggregated, and software-defined 3GPP network implementations, and works next to other standardization entities such as 3GPP.

The goal of Open RAN Alliance is the creation of open networks that are independent from the rigid tie with vendors of networking tools; in order to achieve this goal, it has

fixed technical specifications that allow the service providers to develop software for the control of the mobile network, independently from the hardware.

A feature of the O-RAN Alliance, or simply O-RAN, is the RIC that stands for RAN Intelligent Controller and adds programmability to the RAN. O-RAN is organized in working groups that have their own software, hardware and system components and it determines two types of RICs: the Non-Real-Time RIC and the Near-Real-Time RIC.

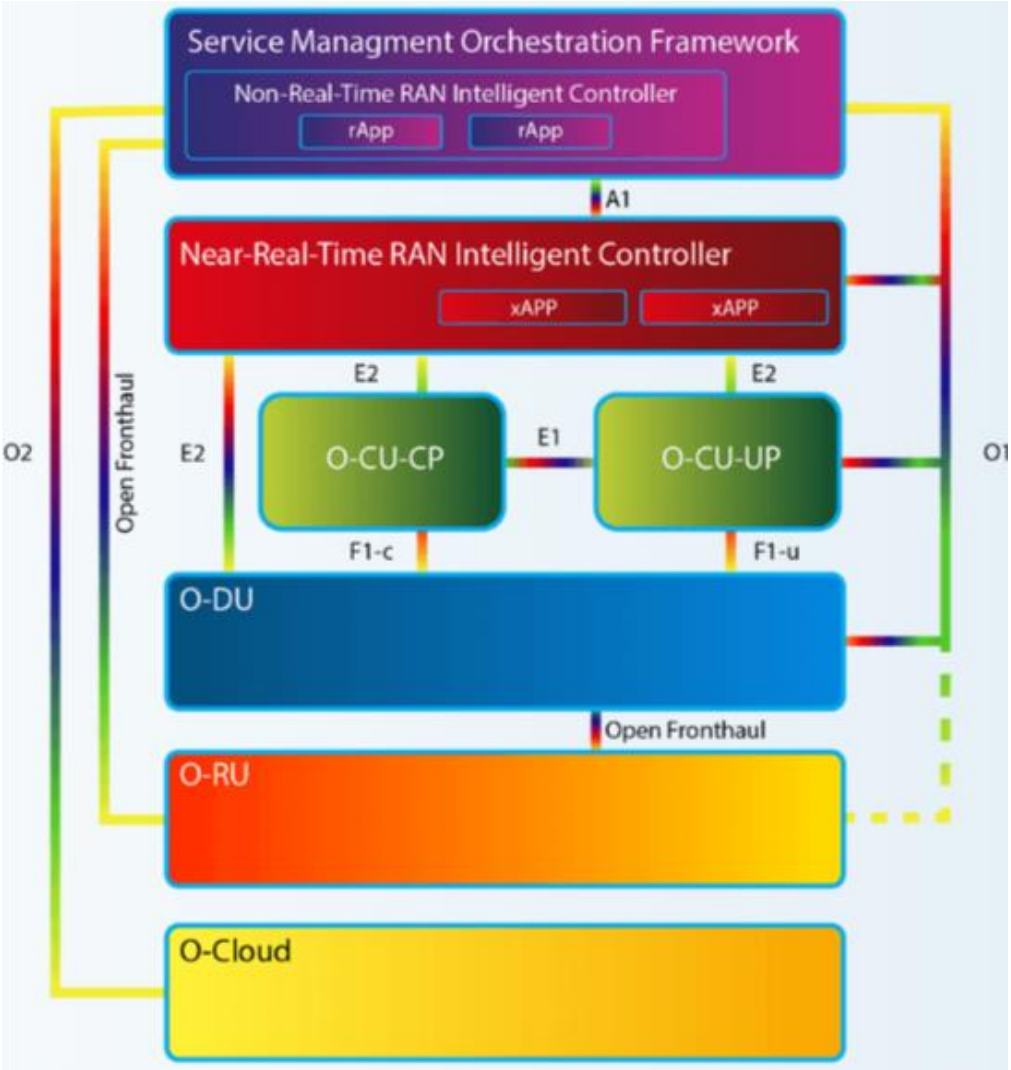


Figure 2.2.4: O-RAN architecture.

The Non-RT RIC controls rApp applications that can handle a latency of one second or higher. The Near-RT RIC takes care of xApp applications that require a latency usually in the range of milliseconds.

The xApps and the rApps are applications delivered by multiple vendors, they provide intelligence for analysis and they need to be interoperable with multi-vendor solutions.

RICs collect network data from the RAN, by means of the application of Artificial Intelligence and/or Machine Learning algorithms, and in pursuance of data collection, a RIC works together with the SMO, which is the abbreviation for Service Management Orchestration.

The basis for the Open RAN architecture is given by Software-Defined Networking (SDN) and Network Functions Virtualization (NFV).

The main principle of Software-defined networking is the physical separation of the control plane from the user plane. The control plane becomes fully software-defined and it handles the decisions for the traffic of data, while the user plane can be placed in hardware of different vendors.

As a result of this separation, the architecture of an SDN accomplishes benefits such as:

- Central management, i.e., global view of the network.
- Fault tolerance.
- Hardware independence.
- Redundancy.
- Agility in implementation.
- Distribution of policy and routing changes.
- Boost of interoperability thanks to open standards incorporation.

SDN makes the control of the network completely programmable, this means that when the network needs a change, the administrators can make the change in the software without the need to manually move a network cable from one switch to another. Besides, SDN enables Private Networks that provide secure networking that can be customize by its owner.

The introduction of new technologies requires more flexibility and therefore a RAN with an architecture more and more software programmable, this is why it is needed the mix of SDN and Network Function Virtualization. Moreover, the combination of SDN and NFV is also needed for a contained latency.

Regardless the fact that SDN, NFV and VNFs share analogous principles and goals, it is not wise to aim to the choice of one of them because it is their union which can give the highest benefit to a network. Through the virtualization of the RAN, software

application infrastructures are created and SDN enables these applications to orchestrate and manage networks. Although SDN can be realized utilizing NFV and NFV can be considered an application scenario of SDN, but the implementation of NFV and SDN are independent.

For what concerns Network Functions Virtualization, it is an architecture that establishes how to execute the functions of SDN on neutral-vendor hardware. NFV is the platform that orchestrates Virtual Network Functions. VNFs are individual network services running on a generic hardware as software virtual machine instances.

Amarisoft removes the hardware dependency and it is able to run all the radio access network on vendor-neutral hardware, this means that 4G and 5G networks can be seen as software upgrades and the RAN can be seen as Virtualized Network Functions.

Blank page.

3 – Literature review

3.1 - Hardware

In pursuance of getting the scenarios that will be illustrated in the Chapter 5.2 of this thesis, many electronic components have been used.

First, the RF wiring in the roof has been set through the usage of coaxial cables connected via N connectors, and a directional coupler of 3dB. The coaxial cables are of brand Cellflex and they are characterized by their contained losses.

In addition to this component, two ways splitters and four ways splitters have been used.

At the beginning, the network has been tested in basic scenarios using three omnidirectional SISO antennas like the one in the following picture.

For more complex scenarios, omnidirectional two port MIMO antennas have been utilized in order to test the network.

Amari Classic Callbox is composed by three PCIe SDR50 cards which have a maximum frequency error less than 2 ppm. This error depends on the SDR hardware and it does not depend on the bandwidth or on the frequency. This error can be fixed by calibrating the PCIe SDR card.

Each SDR card of Amarisoft has a processor and FPGA, it supports LTE FDD and LTE TDD bands and given its 56 MHz bandwidth, it is possible to obtain up to 3 x 20 MHz carrier aggregation.

Figure 3.1.1: Amarisoft SDR card.

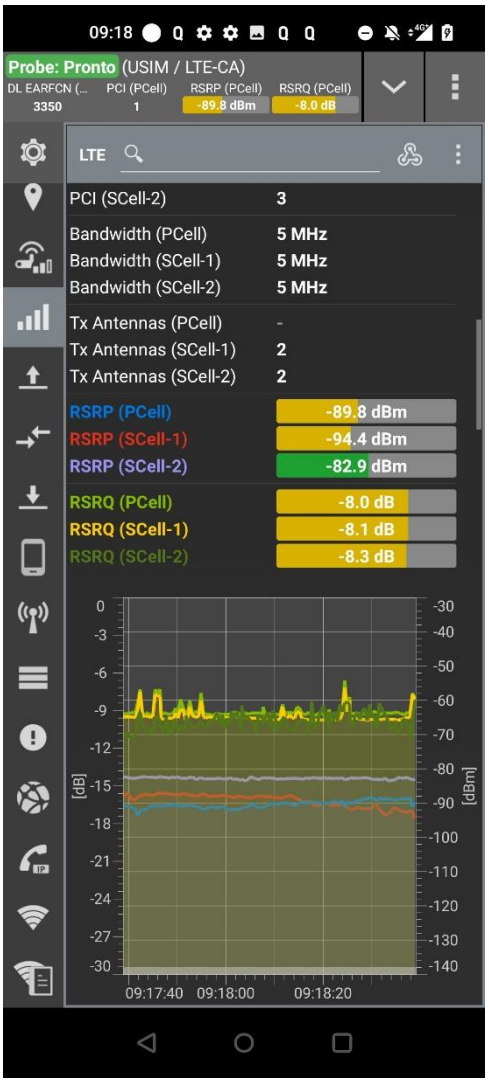
The SDR cards are connected to the splitter through SubMiniature version A (SMA) connectors which are coaxial connectors for high-frequency applications.

3.2 - Software

Of course, the main software used in this thesis is Amarisoft but also applications such as QualiPoc and NEMO have been utilized in order to be able to compare the performances of the different scenarios implemented.

QualiPoc is a smartphone-based tool for the optimization of radio frequency and for voice and data service quality. This application is provided by the Rohde and Schwarz electronics group, it supports numerous protocol layers and it is supported by several commercial smartphones which have Android as their operating system. Moreover, it supports all the technologies for mobile networks, up to the Fifth Generation.

In this multifunctional application there are more than 250 Key Performance Indicators, also called KPIs, that are accessible both in real time and in postprocessing. The test functions that are furnished are for voice, data, video streaming and others and the tests that are performed by the application verify the quality level of services. QualiPoc, therefore, is a very good application to use for the sake of multiple integration tests.



The application, in case of an LTE network, is equipped with technologies features such as:

- Serving and neighbor cells.

- RSSI, RSSP and RSRQ.
- PLMN ID, registration state.
- Serving cell EARFCN.
- Physical cell ID.
- Bandwidth.
- Number of TX antennas.
- Others.

Through the usage three different algorithms, Rohde and Schwarz is able to give its customers detailed analysis of quality. Furthermore, the map monitor of QualiPoc simplifies indoor and outdoor measurements, reducing in this way the complexity of environments that combine outdoor and indoor measurements.

There are various advantages of the QualiPoc application, some of them are listed in the following lines:

- User-friendly interface.
- Cost-efficient.
- Operational efficiency of field engineers.
- QoS and QoE testing feature.
- Strong pocket solution for RF engineers.
- Reliable.
- Real-time interaction.

The reduction of complexity, mentioned before, is a feature that reduces the errors and improve the efficiency in complex settings. Another interesting characteristic is the option “On-device report” which allows the user to obtain a PDF report on the device at the end of a job done.

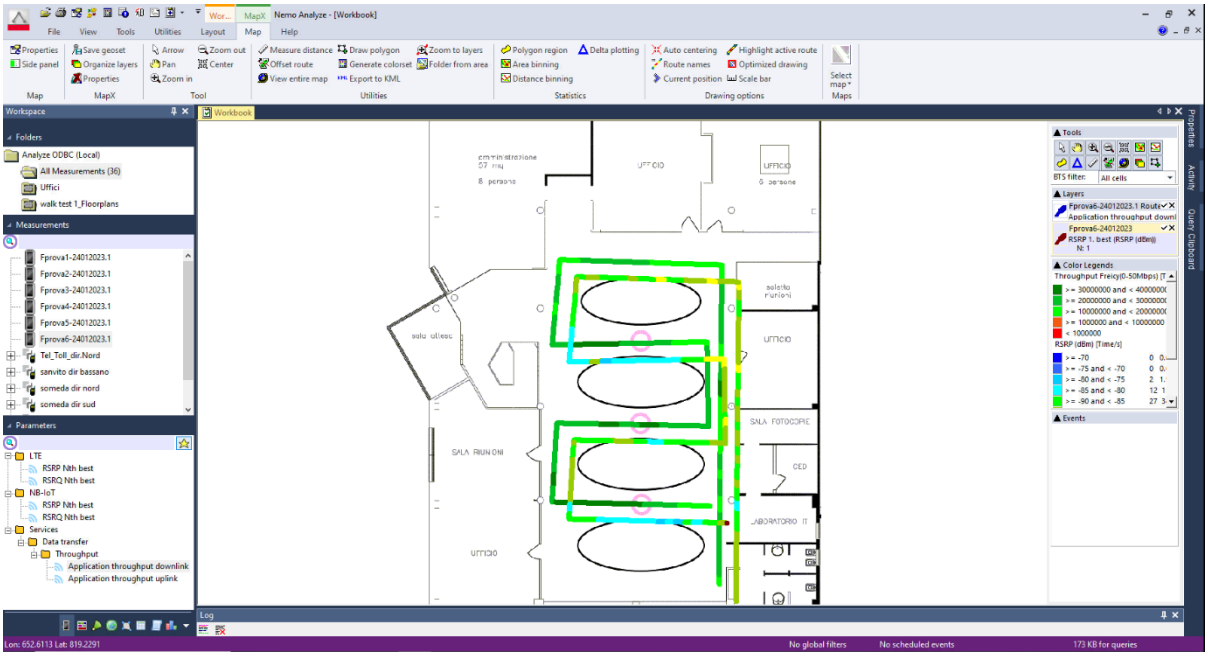
NEMO Analyze is a desktop-based application for analyzing data from Nemo products such as NEMO Handy which is the application used in this thesis.

NEMO is one of the best applications for visualizing the data and customize it, thanks to the inclusion of: maps, grids, line graphs, bar graphs, pie charts, OpenStreetMap, color grids and other engines. In addition to all these characteristics, there are various reporting options, for instance, MS Excel reporting.

The benefits of NEMO Analyze are:

- Highly efficient.
- Fully scalable.
- Automated troubleshooting.
- Latest technologies supported.
- Versatile drive test post-processing.
- Statistical reporting.

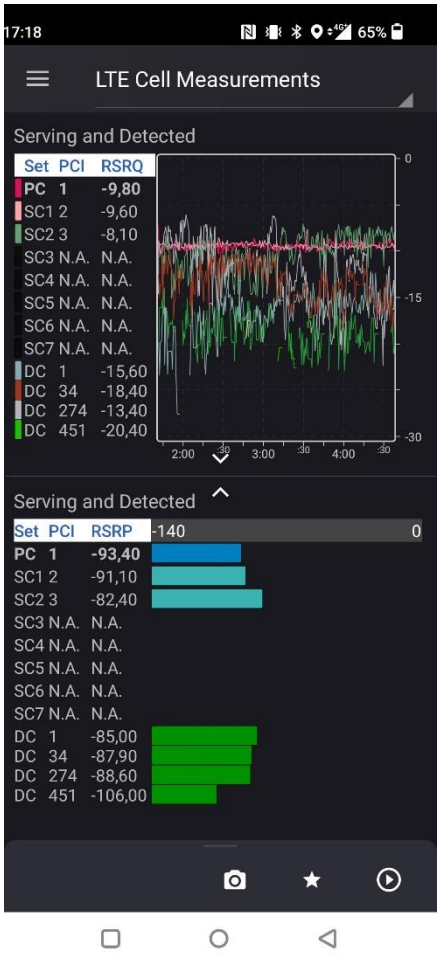
In order to obtain an automated data processing that includes data measurement and automatic results in workbook format, such as PDF, the combination of NEMO Analyze and NEMO tools is needed.

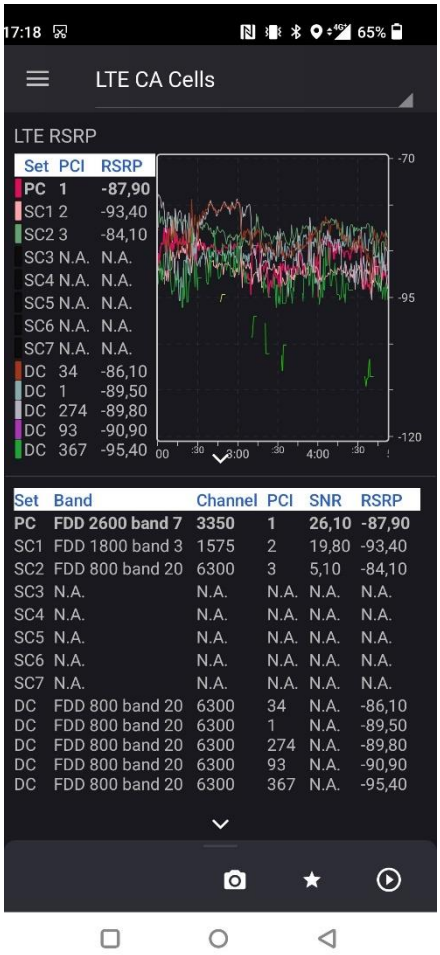


As previously said, both the NEMO application and the NEMO tools support the latest technologies for mobile networks, for instance 5G. This means that a set of KPIs is supported, by NEMO, for 5G New Radio as well. Another characteristic of NEMO Analyze is that it supports measurements executed with Keysight's FieldFox analyzer.

NEMO Analyze is able to implement post-processing analysis and perform statistical reporting based on data from NEMO tools, in this case NEMO Handy.

NEMO Handy is an application for the latest models of smartphones with Android as operating system, it enables wireless measurements and mobile application QoS and QoE.





NEMO Handy supports a wide number of customized scripts that allow to test any application for a set of parameters and it has many advantages:

- Support of the latest wireless technologies up to the Fifth Generation NR.
- Versatile measurement solution.
- Support of indoor measurement.
- QoS measurement.
- QoE measurement.

3.3 - Amarisoft

Amarisoft is a software company which supplies affordable and high-quality solutions to the 4G/5G community, they provide boxes to cover User Equipment (UE), Radio Access Network (RAN) and Core Network (CN) testing, and their unique technology allows to run the full RAN, including the physical layer.

The Amari Callbox is a 3GPP compliant eNodeB/gNodeB and EPC/5GC and it is the perfect solution for testing 5G NSA and SA, LTE and NB-IoT devices. There are four different products offered by AMARI Callbox: Mini, Classic, Advance and Ultimate. The product we will consider in this thesis is the AMARI Callbox Classic.

AMARI Callbox Classic is packaged in a PC and it includes:

- an eNodeB/gNodeB;
- an Evolved Packet Core/5G Core;
- an integrated IP Multimedia Subsystem server;
- a Multimedia Broadcast Multicast Service gateway.

It supports:

- Up to 4x4 MIMO.
- Up to 3x Carrier Aggregation.
- Testing of UEs of category up to CAT10.
- Handover testing.
- Multi-cell configurations.
- Bitrate up to 600 Mbps in DL.
- Bitrate up to 150 Mbps in UL.
- Up to 100 active UEs distributed in the configured cells.

The callbox Classic embeds three Software Defined Radio (SDR) cards and all software components and licenses required in order to emulate Long-Term Evolution networks.

The Amarisoft callbox is delivered with some test SIM cards that are already configured in the EPC database, this means that no additional configuration is required in order to authenticate devices that use these test SIM cards.

The specifications of these SIM cards are:

- IMSI: 001010123456789
- K: 00112233445566778899aabbccddeeff
- sim algo: XOR
- Non programmable USIM card

It is also possible to use commercial sim cards for the sake of testing, in this case you need to add them in the EPC database and to do some further configuration.

Each SDR card supports 20 MHz MIMO 2x2 and it has two receiver connectors and two transmitter connectors plus one GPS connector. RX1/TX1 are the main receive/transmit antenna ports, while RX2/TX2 are the diversity receive/transmit antenna ports. The GPS connector is used to connect an external GPS clock.

When implementing the desired scenario for the network, if RF combiners are used, in order to avoid big insertion losses that downgrade the RF performances, it is needed to terminate unused ports of the combiners with a RF terminator or an absorber, in particular when the frequency used is higher than 2.5 GHz.

The connection to AMARI callbox can be made locally or remotely. In this thesis the connection to the callbox takes place remotely, namely we can access it through a ssh command using root and toor as user and password respectively.

There are some commands that allow to check and keep under control the LTE service:

- `service lte status` -> it checks the status of the LTE service;
- `service lte stop` -> it stops all LTE components;
- `service lte start` -> it starts all LTE components;
- `systemctl disable lte` -> it prevents LTE service to start at boot time;
- `systemctl enable lte` -> it enables LTE service to start at boot time.

The default file used by the LTE service for configuring the eNB/gNB is enb.cfg, this file is available under the /root/enb/config directory. This file sets the eNB/gNB parameters such as frequency (dl_earfcn), cell bandwidth (n_rb_dl), number of layer (n_antenna_dl) and others. The default configuration is one cell FDD SISO in band 7 (LTE) with 5 MHz of bandwidth.

In the enb.cfg file, the parameter rf_port must be set for each cell declared in the cell list called cell_list. By setting the rf_port parameter, the SDR card that will broadcast the cell signal is selected (rf_port:0->SDR0, rf_port:1->SDR1, rf_port:2->SDR2). It is possible to declare up to three cells on different bands. For example, we can have these following scenarios:

- In SISO: 1 SDR card is used and N_ANTENNA_DL: 1
- In 2x2 MIMO: 1 SDR card is used and N_ANTENNA_DL: 2
- In 4x4 MIMO: 2 SDR cards are used and N_ANTENNA_DL: 4

The TX and RX antenna gain values must be customized in order to avoid saturation/BLER if values of antenna gains are too high/low. These values are defined in the RF configuration file that is in the /root/enb/config/rf_driver directory and this file is always included in the enb.cfg file.

The recommended TX/RX gain values change depending on how the DUT (Device Under Test) is connected to the callbox. Recommended values for the wireless case are used: tx_gain: 90.0 and rx_gain: 60.0, both values are in dB.

The LTE Mobility and Management Entity (LTE MME) implementation has a built-in Serving Gateway (SGW), Packet data Network Gateway (PGW), Policy and Charging Rule Function (PCRF), Home Subscriber Server (HSS) and Equipment Identity Register (EIR). It also includes a New Radio 5G Core (NR 5GC) with a built-in Access and Mobility Management Function (AMF), Authentication Server Function (AUSF), Session Management Function (SMF), User Plane Function (UPF).

The default file used for configuring the core network is mme.cfg, this file is available under the /root/mme/config directory and it sets parameters such as PLMN ID, network name, PDN list, APN name and others.

The default PLMN ID of the callbox is 00101, where 001 is the MCC and 01 is the MNC. If a commercial sim card is used, the PLMN value must be updated with the MCC/MNC values of the commercial SIM card in the mme.cfg file and the new PLMN value must be added also in the plmn_list array in the /root/enb/config/enb.cfg file.

If two AMARI test SIM cards are used, the MME allows each UE to be attached to the eNB with the same IMSI, distinguishing them with their IMEI. This is possible thanks to the line “multi_sim: true,” inside the ue_db-ims.cfg file.

The IMEI number can be found from the screen of the eNB by digiting the command “ue”, one of the outputs of this command is the IMEISV number, which stands for IMEI Software Version and it is possible to extract IMEI from IMEISV in this way: IMEI = IMEISV – last two digits + 0.

If more than two AMARI test sim cards are used, their IMEI need to be declared in the EPC database called HSS in this way:

```
impu: [ "001010123456789",
        {impu: "tel:0801", imei:"866929050364853"},
        {impu: "tel:0802", imei:"866929050363731"},
        {impu: "tel:0803", imei:"350117931057059"}, ],
```

If a commercial sim card is used, it will have different IMSI so you need to declare it in the EPC database (HSS), in order to do so, you need to know the secret key values and add these values in the /root/mme/config/ue_db-ims.cfg file, inside the ue_db array in this way:

```
{  sim algo: "milenage",
imsi: "001010000000001",
opc: "000102030405060708090A0B0C0D0E0F",
amf: 0x9001,
sqn: "000000000000",
K: "00112233445566778899AABBCCDDEEFF", },
```

The secret keys parameters are mandatory because both the device and the network need to authenticate each other, nevertheless most of the times these values are unknown due to their confidentiality. In this case, the solution consists in disabling the authentication using one of these two methods:

1- If the device accepts the use of the EIA0 encryption algorithm then in the mme configuration file the command authentication_mode: “skip” must be added.
2- If the command at the UE side to skip both authentication and security control procedures is known then in the mme configuration file the commands authentication_mode: “skip” and skip_smc_proc: true must be added and in the eNB configuration file the command skip_smc_proc: true must be added.

Even though the MME and eNB configuration files include the commands just mentioned, disabling these procedures might be rejected by the UE unless it is in a specific test mode. The reason behind the rejection is that it is not allowed by 3GPP specifications to disable authentication and security operations.

The device must be in a very specific test mode in order to be able to skip the authentication and security mechanisms, otherwise it will not be possible to use a commercial SIM card. The “specific test mode” is designed only by the modem manufacturer that is able to allow this non-3GPP compliant behavior, for this reason this thesis has been written taking in consideration two SIM cards provided by Amarisoft.

MME is configured by default with four APNs: Default, Internet, IMS, SOS. For connecting the UE to the callbox, one of these APNs must be configured at the UE side. If the DUT must be under a specific APN, the modification can be done in the mme.cfg file, creating a new entry in the pdn_list array and doing the restart of the LTE service.

AMARI callbox includes an IMS server that can be used to run IMS tests such as VoLTE or SMS over IMS. In the following chapters the two of them will be tested.

For the sake of internet connection, the “internet” APN must be configured at the UE side to match the callbox setting. At the UE side, the modifications to be done are:

- Mobile data must be ON.
- Data roaming must be ON.
- VoLTE call must be ON, for VoLTE tests.
- “Internet” APN must be created, saved and selected.

It is possible to create a new APN in the setting menu, going to Settings/Wi-Fi&Network/SIM&Network/SIM 1 (USIM)/APNs:

- Name = “Internet”.
- APN = “internet”.
- APN type = “internet,default”.

Everytime there is a change of configuration in any file, the command “service lte restart” is needed in order to update the configurations that have been made.

To monitor the callbox components (eNB, MME, IMS or MBMSGW), it is possible to access these software components using the screen command “*screen –x lte”,* this command allows the access to different monitor components.

Each component monitor is inside a window and it is possible to switch from one window to another with the following command:

```
ctrl+a <space> or ctrl+a <window_index>
```

Where the window_index can be:

- 0 for MME;
- 1 for eNB;
- 2 for MBMSGW;
- 3 for IMS.

Each screen offers a list of commands that can be utilized and it is possible to list them using the command “help”. For instance, in the eNB screen, there are useful commands to provide key information about the Uplink and Downlink transfer(“t”), to provide key information about the CPU load (“t cpu”), etc. To exit the screen mode the command to use is: “ctrl+a d”.

In this thesis, the monitors will be occasionally denoted as:

- (mme) for the MME;
- (enb) for the eNodeB;
- (mbmsgw) for the MBMS gateway;
- (ims) for the IMS.

All the components of the callbox generate log files under the tmp directory: enb0.log, ims.log, mbmsgw.log, mme.log. These log files can be utilized for analysis and debugging. Amarisoft LTE software logs can also be analyzed through the LTE web interface that allows to get real time information from the system.

In the following chapters, the first thing done is the implementation of an LTE network to study the attachment procedure, and to test different parameters in different scenarios with different number of cells.

3.4 - AMARI Callbox Classic Architecture

Each component of AMARI callbox is run by different configuration files, the following graph can be used to understand how these files are interconnected.

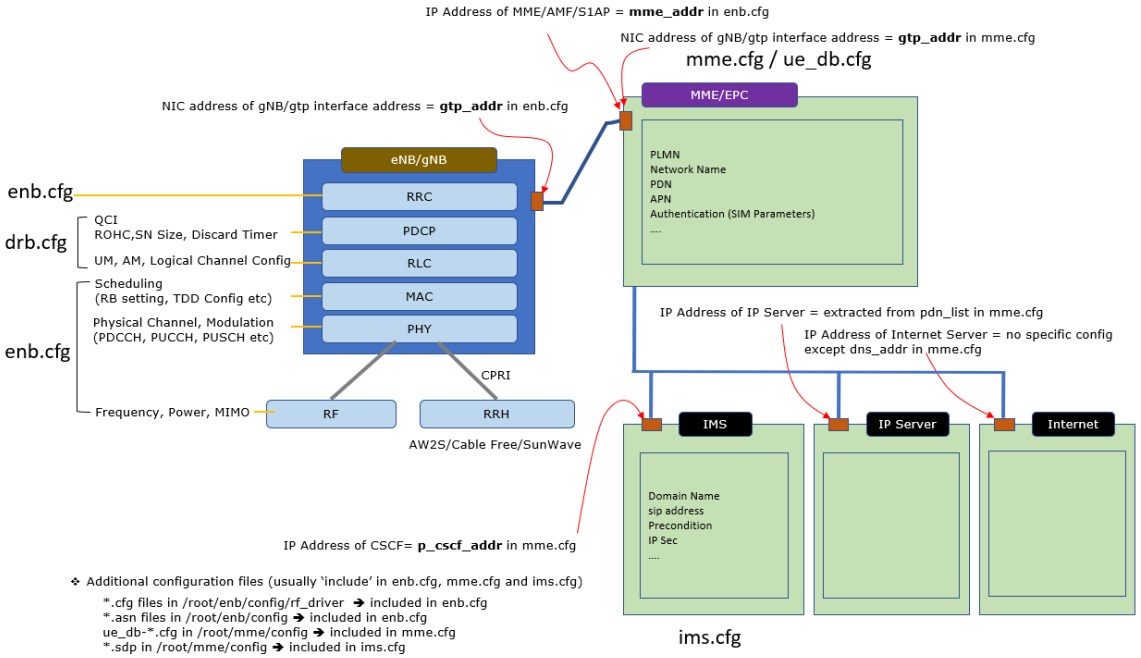


Figure 3.4.1: AMARI Callbox architecture.

From this picture it is visible in which directories you can find the configuration files. For instance, in the /root/enb/config directory there are the following files:

- enb.cfg file;
- drb.cfg file;
- rf_driver directory with its config.cfg file;
- sdr directory with its config.cfg file.

On the other side, the mme directory includes the configuration files of the mme, ims, database of UEs and others. The enb.cfg file encompasses the definition of number of cells, bandwith, number of antennas, rf ports used, list of cells with their neighbour cells, PUCCH dedicated configuration, SRS dedicated configuration, and others.

The Data Radio Bearer configuration file is included in the enb.cfg file and it includes the DBR configuration for each QoS Class Identifier (QCI). There must be at least one definition for the default QCI, which is equal to nine and it is used for internet traffic.

The QCI value is assigned to each UE in the attachment procedure in one of the NAS messages exchanged between the end device and the callbox. It ensures that the traffic is allocated with the right Quality of Service.

Different QoS requirements means different QCI values. In the image below there are the most common QCI values where GBR stands for Guaranteed Bit Rate.

QCI	Resource Type	Priority	Packet Delay Budget (NOTE 1)	Packet Error Loss Rate (NOTE 2)	Example Services
1 (NOTE 3)	GBR	2	100 ms	10^{-2}	Conversational Voice
2 (NOTE 3)		4	150 ms	10^{-3}	Conversational Video (Live Streaming)
3 (NOTE 3)		3	50 ms	10^{-3}	Real Time Gaming
4 (NOTE 3)		5	300 ms	10^{-6}	Non-Conversational Video (Buffered Streaming)
5 (NOTE 3)	Non-GBR	1	100 ms	10^{-6}	IMS Signalling
6 (NOTE 4)		6	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7 (NOTE 3)		7	100 ms	10^{-3}	Voice, Video (Live Streaming) Interactive Gaming
8 (NOTE 5)		8	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9 (NOTE 6)		9			

Figure 3.4.2: QCIs and their services.

In the ./rf_driver/config.cfg file it is possible to define the SDR cards used and it is possible to set tx_gain and rx_gain. In this file it is present the parameter rx_latency and this file it is linked to the ./sdr/config.cfg file. In the /root/mme/config directory, there are the following files:

- mme.cfg;
- ims.cfg;
- ue_db-ims.cfg.

Amarisoft LTEIMS server is an IMS standalone server that supports dedicated bearer using Rx interface. It comprises:

- HSS, Home Subscriber Server.
- CSCF, Call Session Control Function.

HSS is the server in which all the subscribers are registered and it is used to authenticate the UEs. CSCF is the node handling the Session Initiation Protocol (SIP) signaling invoking applications and controlling the media path.

The graph beneath shows how the different components of Amarisoft LTEIMS server are interconnected among them and how Amarisoft LTEIMS server is connected to Amarisoft LTEMME. In particular: Rx is the interface between LTEIMS server and

LTEMME; Cx is the interface between I/S-CSCF and HSS and Mw is the interface among S/I/P-CSCF.

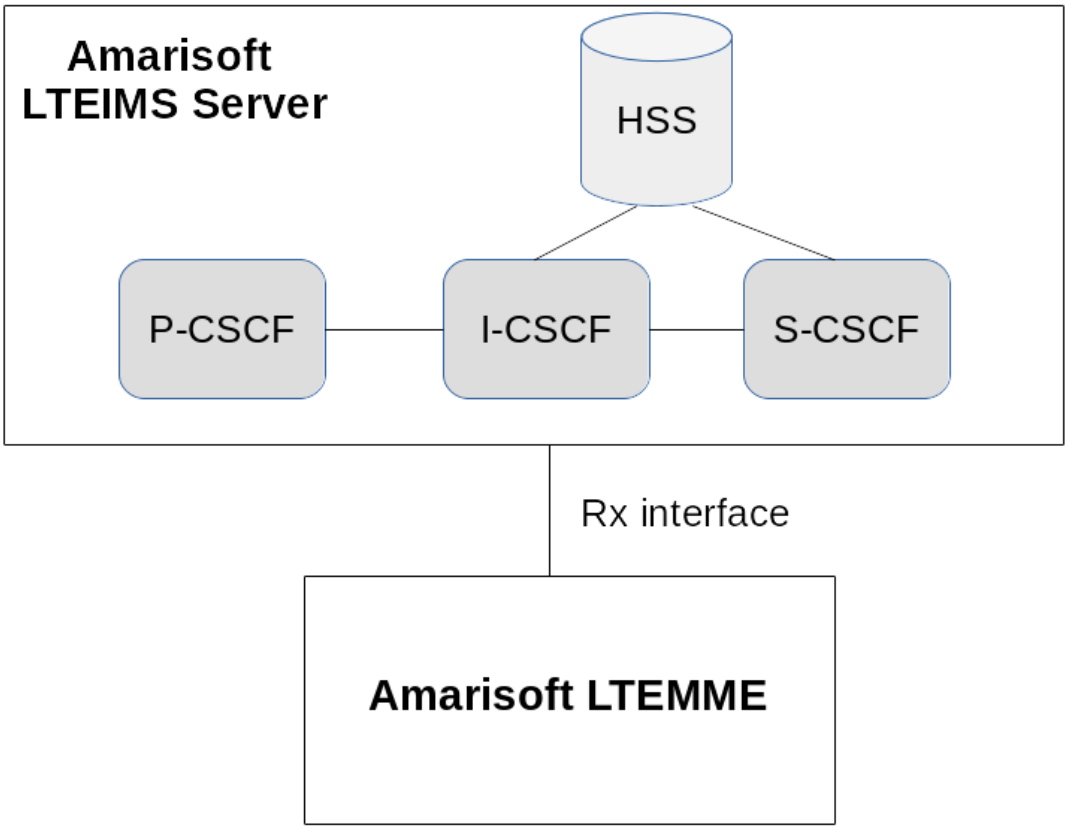


Figure 3.4.3: Connections between LTEMME and LTEIMS.

It is perceptible that the CSCF block is further divided in three different entities:

- P-CSCF, Proxy Call Session Control Function.
- I-CSCF, Interrogating Call Session Control Function.
- S-CSCF, Serving Call Session Control Function.

P-CSCF is the first point of contact for a user and the P-CSCF assigned to a user does not change for the duration of the registration.

I-CSCF is the entry contact within an operator network and it forwards SIP request or response to the S-CSCF.

S-CSCF is a SIP server, it is responsible for the registration process and downloading user information from the HSS.

Amarisoft LTEIMS server cannot be utilized for voice calls between a UE registered in Amarisoft network and an external UE in another network.

LTEIMS furnishes the framework for multimedia services that are IP based in a mobile network. IMS uses SIP (Session Initiation Protocol) protocol to arrange and control voice calls, video calls and instant messaging.

In order to allow the request of access to the IMS domain from the user, an IP multimedia Services Identity Module (ISIM) application is added to the Universal Integrated Circuit Card (UICC). The ISIM application consists of:

- a private ID called IMPI (IP Multimedia Private Identity): this ID is usually given by the home operator and it is used to identify the subscription of the IMS user. The main role of IMPI is to support the authentication procedure during in the phase of registration of the attachment;
- one or more public IDs called IMPU (IP Multimedia Public Identity): this ID is used for user-to-user communication. The IMPU is utilized for message routing for IMS session-based SIP messages;
- a home Domain: the IMS routes the users' registration request to the Home IMS Network thanks to the instruments provided by the home Domain.

If there is no ISIM application, it is possible to obtain it from the knowledge of the IMSI, and therefore from the awareness of MCC and MNC, in this way: ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org. Also, in case there is no ISIM, the private ID IMPI can be derived knowing MCC and MNC in this way: IMSI@ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org. In the same way, it is possible either to use random chosen IMPUs or to derive the public IDs in this way: sip:IMSI@ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org. An example can be:

IMSI = 234150999999999
MCC = 234
MNC = 15
Domain Name = ims.mnc015.mcc234.3gppnetwork.org
IMPI = 234150999999999@ims.mnc015.mcc234.3gppnetwork.org
IMPU = sip:234150999999999@ims.mnc015.mcc234.3gppnetwork.org

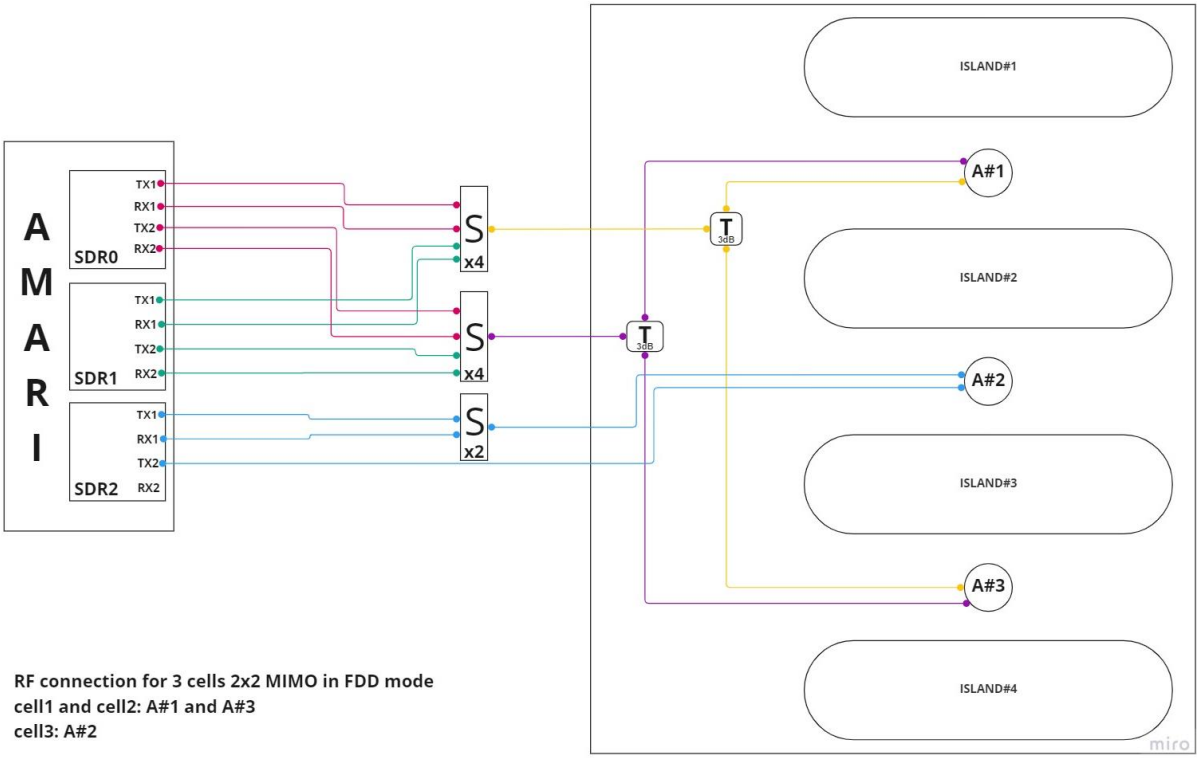
Once the eNB and the MME have been properly configurated, it is needed to restart the LTE service through the command "service lte restart".

The Off-The-Shelf package is the component that allows to perform the initialization of the LTE service at each boot of the system. The ots.cfg file encloses all the components that the LTE service automatically initialize when booting the system. It is needed to define all the components of the LTE service, if there are multiple eNBs, multiple MMEs or multiple IMSs, they will be all present in the ots.cfg file.

4 – Implementation of LTE network in different scenarios

4.1 - RF connection diagrams

In this thesis different scenarios have been tested and in order to be able to evaluate them it was necessary to properly set up the RF connections. The electrical wiring has been distributed for the scenario with the highest complexity in the direction of being able to reach all the desired configurations afterwards. The most complex scenario taken into account in this thesis is the one with three cells 2x2 MIMO in FDD mode, where there are two cells in antenna number one and antenna number three, and one cell in antenna number two. This means that antenna number one and three have two sectors, while antenna number two has only one sector.

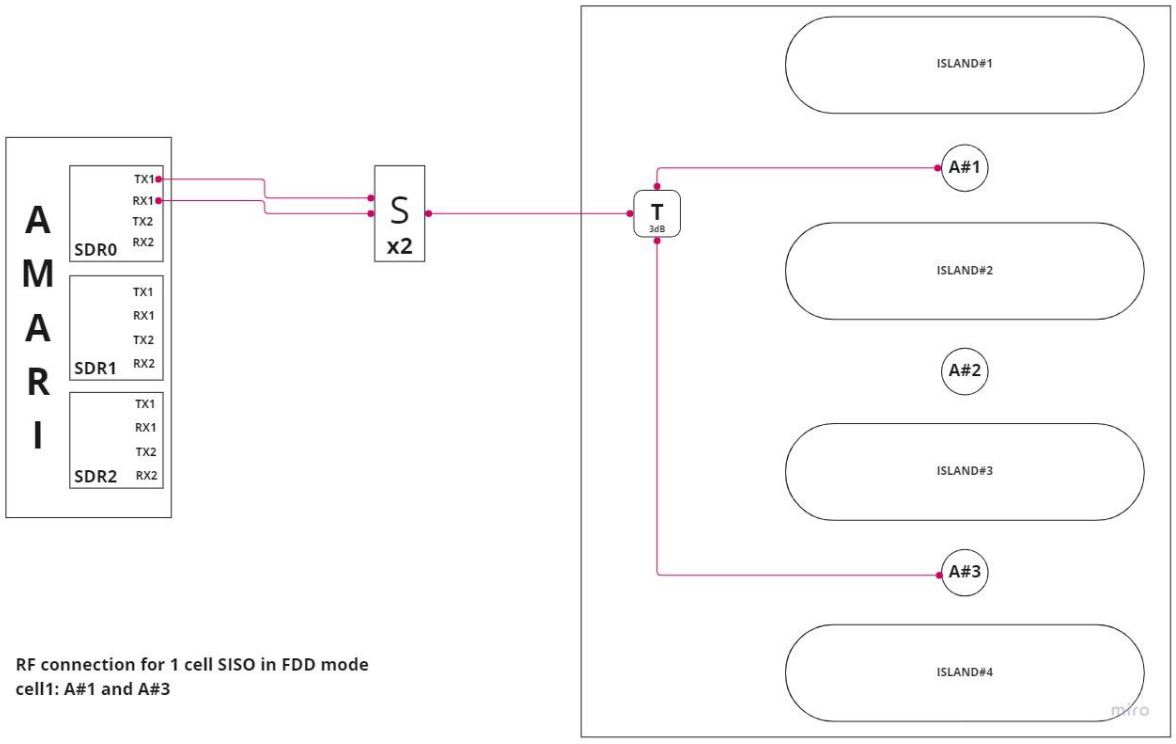


RF connection for 3 cells 2x2 MIMO in FDD mode
cell1 and cell2: A#1 and A#3
cell3: A#2

It is important to take into account some factors. First, in order to change from TDD mode to FDD mode, and viceversa, it is sufficient to include the command “rx_antenna: auto” in the file /rf_driver/config.cfg that it is placed in the /root/enb/config directory. Then, in the scenarios with more than one cell, the RX2 port connection is not needed because, up to the version used in this thesis of Amarisoft (Release of 16-09-2022), uplink MIMO is not supported. Starting from the RF connection for three

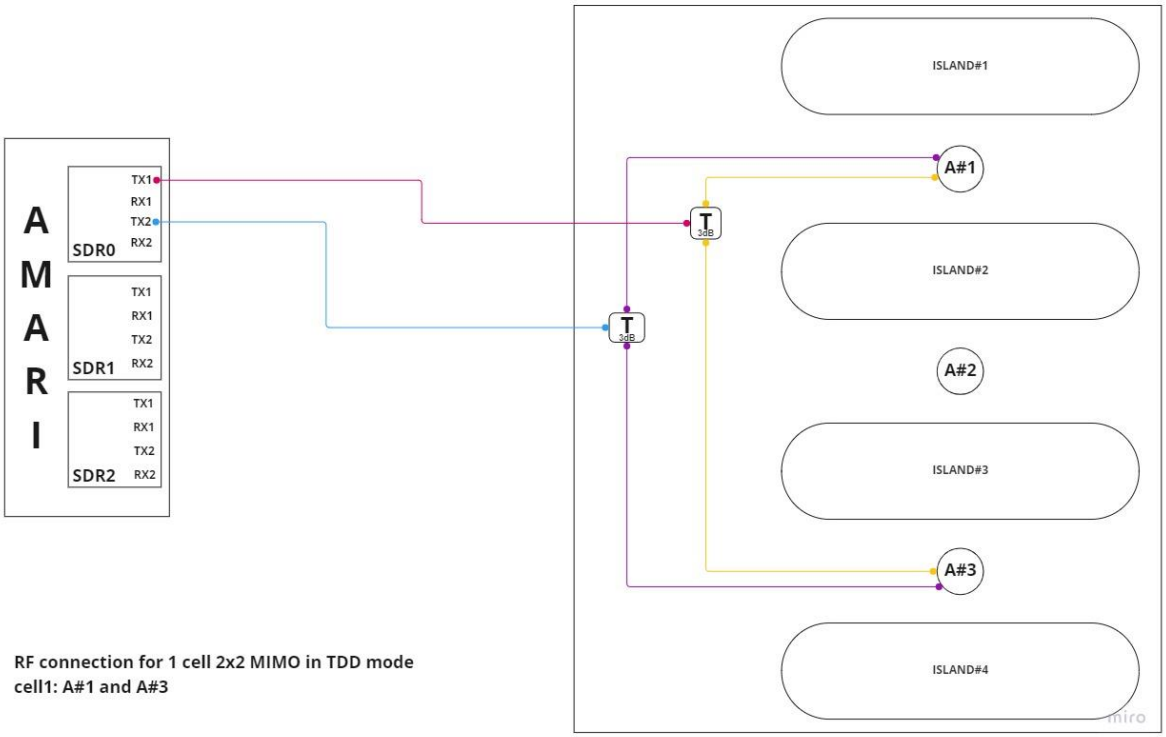
cells 2x2 MIMO in FDD mode, it is possible to derive the other scenarios, that will be discussed in the coming chapters:

- One cell SISO in FDD mode.



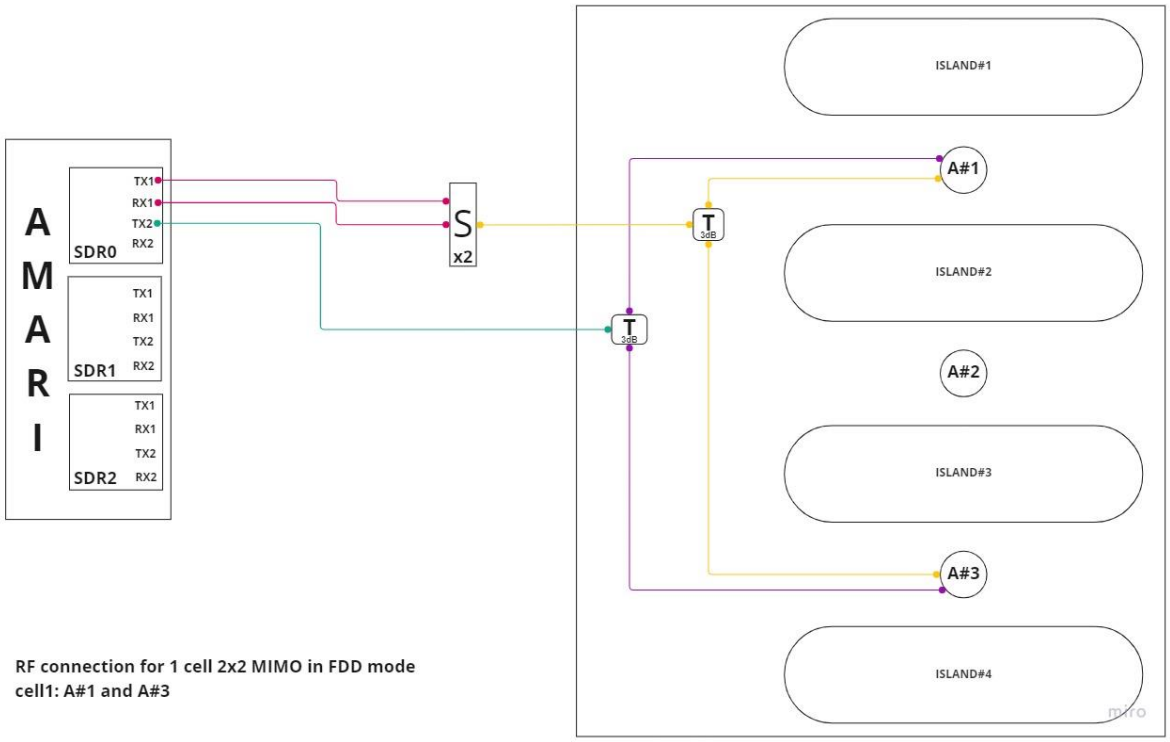
RF connection for 1 cell SISO in FDD mode
cell1: A#1 and A#3

- One cell 2x2 MIMO in TDD mode.



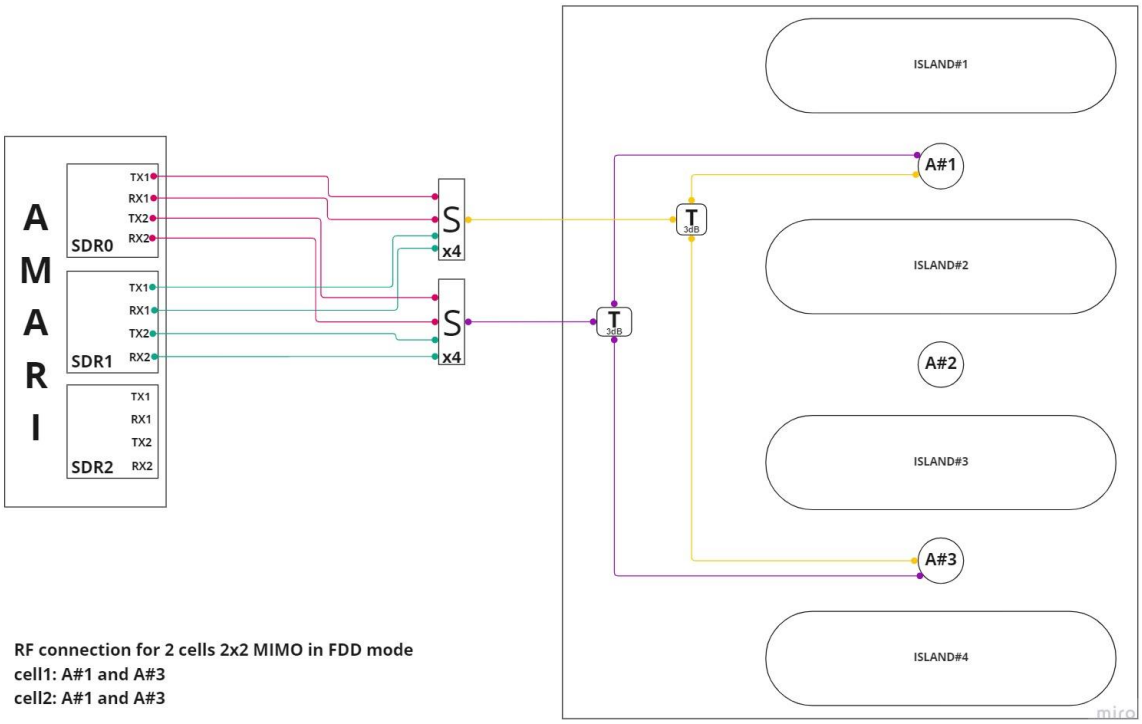
RF connection for 1 cell 2x2 MIMO in TDD mode
cell1: A#1 and A#3

- One cell 2x2 MIMO in FDD mode.



RF connection for 1 cell 2x2 MIMO in FDD mode
cell1: A#1 and A#3

- Two cells 2x2 MIMO in FDD mode.



RF connection for 2 cells 2x2 MIMO in FDD mode
cell1: A#1 and A#3
cell2: A#1 and A#3

4.2 - Attachment procedure

After being switched on, the attachment is the first thing the device executes, without this operation the DUT would not be able to benefit the services from the desired network. An example of a theoretical attachment procedure is illustrated in the book of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan:

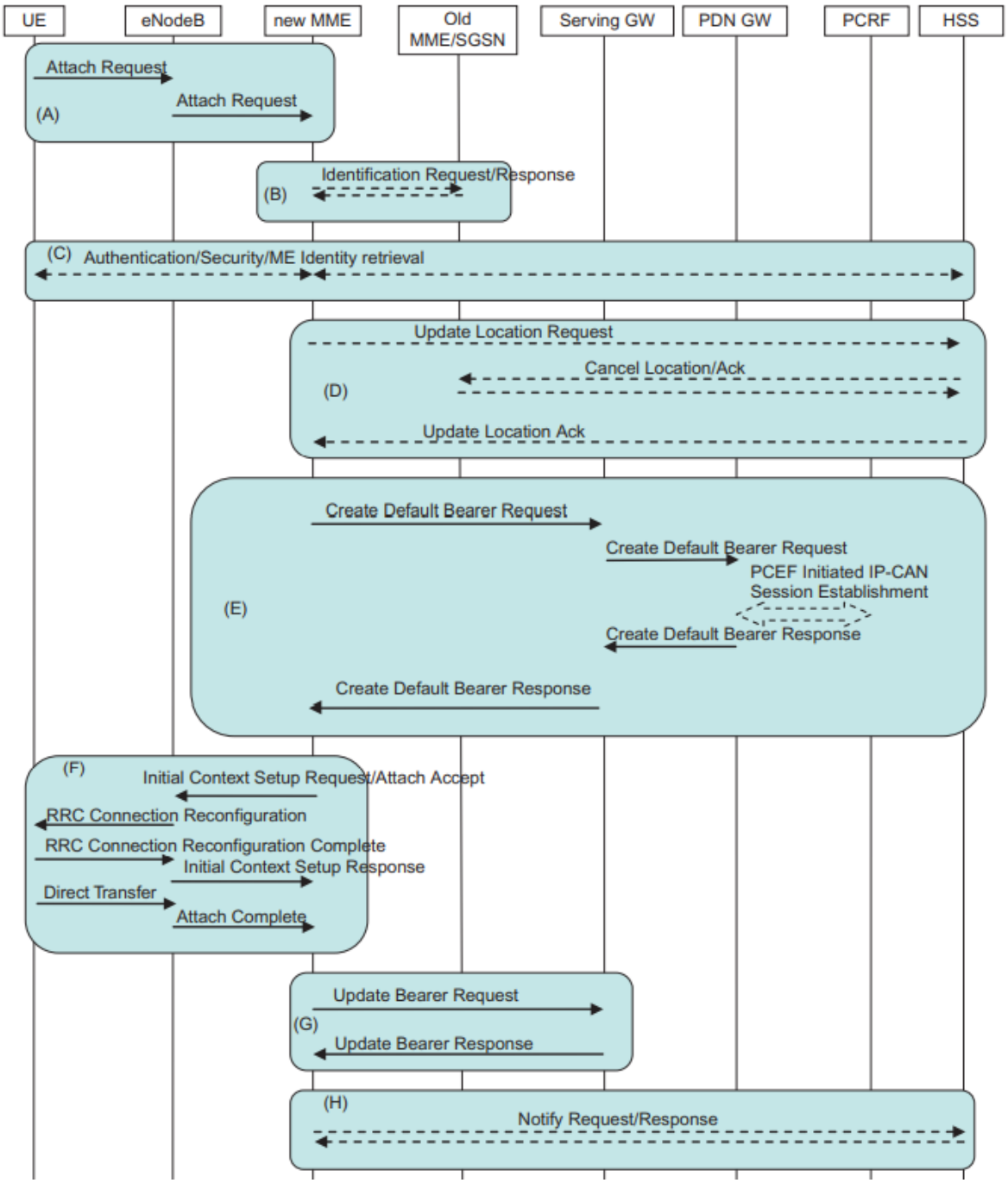


Figure 4.2.1: Theoretical attachment procedure.

Only after the attachment is completed, the device is able to receive services from the network. The attachment for E-UTRAN includes the establishment of a default EPS bearer, in this way the IP connectivity is always guaranteed.

In the interest of understanding the messages exchanged between the DUT and the Amari Callbox, it is important to keep in mind the protocol stacks of the Control Plane and the User Plane for LTE networks. The table that represents the User Plane protocol stack is:

<table>
<tr><td>APPLICATION</td><td colspan="3"></td><td></td><td>APPLICATION</td></tr>
<tr><td>IP</td><td colspan="3"></td><td>IP</td><td>IP</td></tr>
<tr><td>PDCP</td><td>PDCP</td><td>GTP-U</td><td>GTP-U</td><td>GTP-U</td><td rowspan="3">L2</td></tr>
<tr><td rowspan="2">RLC</td><td rowspan="2">RLC</td><td>UDP</td><td>UDP</td><td>UDP</td></tr>
<tr><td>IP</td><td>IP</td><td>IP</td></tr>
<tr><td>MAC</td><td>MAC</td><td>L2</td><td>L2</td><td>L2</td><td></td></tr>
<tr><td>PHY</td><td>PHY</td><td>L1</td><td>L1</td><td>L1</td><td>L1</td></tr>
<tr><td>DUT</td><td colspan="2">eNB</td><td>S-GW</td><td>P-GW</td><td>PDN</td></tr>
</table>

The table that represents the Control Plane protocol stack is:

<table>
<tr><td>NAS</td><td colspan="2"></td><td>NAS</td></tr>
<tr><td>RRC</td><td>RRC</td><td>S1-AP</td><td>S1-AP</td></tr>
<tr><td>PDCP</td><td>PDCP</td><td>SCTP</td><td>SCTP</td></tr>
<tr><td>RLC</td><td>RLC</td><td>IP</td><td>IP</td></tr>
<tr><td>MAC</td><td>MAC</td><td>L2</td><td>L2</td></tr>
<tr><td>PHY</td><td>PHY</td><td>L1</td><td>L1</td></tr>
<tr><td>DUT</td><td colspan="2">eNodeB</td><td>MME</td></tr>
</table>

From these tables, it is observable that in order to attach the DUT to the desired network, the device needs to send a connection request to the eNB and this request will be forwarded to the MME using the S1-AP protocol. Amarisoft allows the user to see the transfer of messages in a user-friendly interface. First of all, the eNB receives a RRC connection request and sends back a connection setup message.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+7.248	RRC			65		1	CCCH	RRC connection request
	RRC			65		1	CCCH	RRC connection setup

At the core network side, the MME receives both the initial UE message and the attach request from the eNB, it reads the information about the device, which includes the ciphering algorithms supported by the UE that is important information for the security procedure, and finally it answers with a “ESM information request” message.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.099		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Initial UE message
		NAS		148 (101)	001010123456789		EMM	Attach request
		NAS		148 (101)	001010123456789			EPS encryption caps=0xf0 integrity caps=0x70
		NAS		148 (101)	001010123456789		ESM	ESM information request
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Downlink NAS transport

Once the RRC connection setup is completed, the DUT is able to send the attach request to the MME via the eNB of course. Furthermore, the core network and the radio access network exchange some information about the UE.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	RRC			65		1	DCCH	RRC connection setup complete
	NAS			65			EMM	Attach request
	S1AP							127.0.1.100:36412 Initial UE message
+0.001	S1AP			148				127.0.1.100:36412 Downlink NAS transport

Afterwards, the MME sends an ESM information request, where information about the used APNs is placed, and waits for the ESM information response.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	NAS			65			ESM	ESM information request
	RRC			65		1	DCCH	DL information transfer
+0.023	RRC			65		1	DCCH	UL information transfer
	NAS			65			ESM	ESM information response

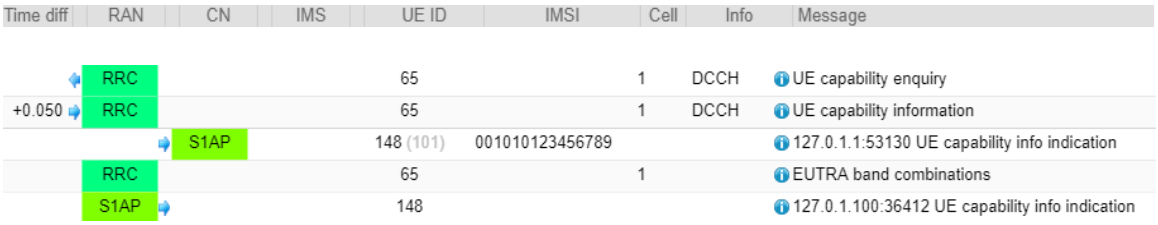
Only after the ESM information response is received by the MME, this entity is able to answer with the attach accept message.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	S1AP			148				127.0.1.100:36412 Uplink NAS transport
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Uplink NAS transport
		NAS		148 (101)	001010123456789		ESM	ESM information response
		NAS		148 (101)	001010123456789		EMM	Attach accept
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Initial context setup request
+0.001	S1AP			148				127.0.1.100:36412 Initial context setup request

Subsequently to the attach accept message, the RRC messages for the security procedure are transferred, in these messages it is possible to find information about the ciphering algorithm used by the device under test. The ciphering algorithm used is important in case the authentication phase needs to be skipped for commercial SIM cards with secret values unknown.

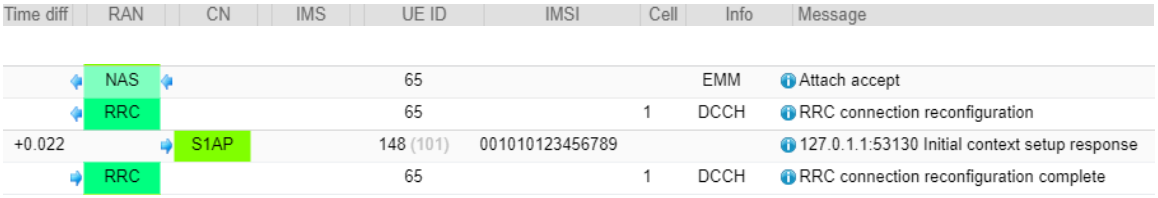
Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	RRC			65		1	DCCH	Security mode command
+0.027	RRC			65		1	DCCH	Security mode complete

At this point, the eNB asks the UE for its capability information and the UE answers with the RRC message labelled “UE capability information” in which it is possible to find the category of the device. In this thesis the devices utilized are of category 4. This category will be useful when testing the throughput. Furthermore, the eNB sends the UE capability information to the MME and, after processing the “EUTRA band combinations” message, it sends a “UE capability information indication” as well.



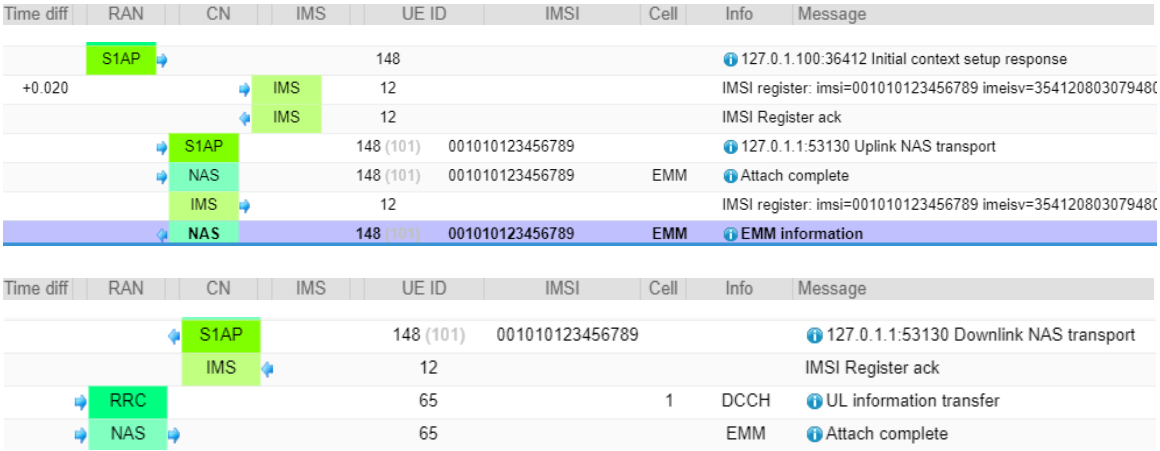
Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	RRC			65		1	DCCH	UE capability enquiry
+0.050	RRC			65		1	DCCH	UE capability information
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 UE capability info indication
	RRC			65		1		EUTRA band combinations
	S1AP			148				127.0.1.100:36412 UE capability info indication

Finally, the eNB receives the attach accept from the MME and it forwards it to the UE. After the acceptance, the RRC connection reconfiguration takes place.



Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	NAS			65			EMM	Attach accept
	RRC			65		1	DCCH	RRC connection reconfiguration
+0.022		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Initial context setup response
	RRC			65		1	DCCH	RRC connection reconfiguration complete

Later, the IMSI registration in the LTEIMS takes place and after some exchange of EMM information and IMSI registration Acknowledgement, the attach complete message is finally exchanged between the DUT and the MME blocks of Amari Callbox.



Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	S1AP			148				127.0.1.100:36412 Initial context setup response
+0.020			IMS	12				IMSI register: imsi=001010123456789 imeisv=354120803079480
			IMS	12				IMSI Register ack
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Uplink NAS transport
		NAS		148 (101)	001010123456789		EMM	Attach complete
		IMS		12				IMSI register: imsi=001010123456789 imeisv=354120803079480
		NAS		**148 (101)**	**001010123456789**	**EMM**		**EMM information**

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Downlink NAS transport
		IMS		12				IMSI Register ack
	RRC			65		1	DCCH	UL information transfer
	NAS			65			EMM	Attach complete

After the attachment procedure is completed, the PDN connectivity takes place through the request for activation of default EPS bearer, which is the bearer between the UE and the PDN gateway, e.g., it is the bearer for accessing internet.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
	S1AP			148				127.0.1.100:36412 Uplink NAS transport
+0.001	S1AP			148				127.0.1.100:36412 Downlink NAS transport
	NAS			65			EMM	EMM information
	RRC			65		1	DCCH	DL information transfer
+0.287		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Uplink NAS transport
		NAS		148 (101)	001010123456789		ESM	PDN connectivity request
		NAS		148 (101)	001010123456789		ESM	Activate default EPS bearer context request
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 E-RAB setup request
	RRC			65		1	DCCH	UL information transfer
	NAS			65			ESM	PDN connectivity request
	S1AP			148				127.0.1.100:36412 Uplink NAS transport
+0.001	S1AP			148				127.0.1.100:36412 E-RAB setup request
	NAS			65			ESM	Activate default EPS bearer context request
	RRC			65		1	DCCH	RRC connection reconfiguration

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.039		S1AP		148 (101)	001010123456789			127.0.1.1:53130 E-RAB setup response
		S1AP		148 (101)	001010123456789			127.0.1.1:53130 Uplink NAS transport
		NAS		148 (101)	001010123456789		ESM	Activate default EPS bearer context accept
	RRC			65		1	DCCH	RRC connection reconfiguration complete
	S1AP			148				127.0.1.100:36412 E-RAB setup response
	RRC			65		1	DCCH	UL information transfer
	NAS			65			ESM	Activate default EPS bearer context accept
	S1AP			148				127.0.1.100:36412 Uplink NAS transport

After the acceptance of the activation of the default EPS bearer starts the SIP/IMS registration so that the UE is able to communicate with the LTEIMS other than being able to access internet.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.380		IMS						Connection from [2001:468:3000:1:2001:468:3000:1]:46819
+0.018	CX							127.0.0.1:34954 Multimedia-Auth-Request
	CX							127.0.0.1:34954 Multimedia-Auth-Answer
		SIP	95 (50)	001010123456789	REGISTER			sip:test.3gpp.com SIP/2.0 from [2001:468:3000:1:2001:468:3000:1]:46819
		SIP	95 (50)	001010123456789	SIP/2.0			401 Unauthorized to [2001:468:3000:1:2001:468:3000:1]:5060
+0.074		IMS						Socket closed by remote host: [2001:468:3000:1:2001:468:3000:1]:46819
+0.120		IMS						Connection from [2001:468:3000:1:2001:468:3000:1]:6901
+0.019		SIP	96 (50)	001010123456789	REGISTER			sip:test.3gpp.com SIP/2.0 from [2001:468:3000:1:2001:468:3000:1]:6901
		IMS	96 (50)	001010123456789				001010123456789 authenticated
		IMS	97 (50)	001010123456789				Register 001010123456789@2001:468:3000:1:2001:468:3000:1:6900
		SIP	96 (50)	001010123456789	SIP/2.0			200 OK to [2001:468:3000:1:2001:468:3000:1]:6900
+0.099		SIP	98 (50)	001010123456789	SUBSCRIBE			sip:001010123456789@ims.mnc001.mcc001.3gppnetwork.org SIP/2.0 from [2001:468:3000:1:2001:468:3000:1]:690
		SIP	98 (50)	001010123456789	SIP/2.0			200 OK to [2001:468:3000:1:2001:468:3000:1]:6900
		IMS						Connect to [2001:468:3000:1:2001:468:3000:1]:6900, ipsec
		SIP	98 (50)	001010123456789	NOTIFY			sip:001010123456789@[2001:468:3000:1:2001:468:3000:1]:6900 SIP/2.0 to [2001:468:3000:1:2001:468:3000:1]:69
+0.087		SIP	98 (50)	001010123456789	SIP/2.0			200 OK from [2001:468:3000:1:2001:468:3000:1]:6900

4.3 - SMS over IMS

In order to be able to send SMS over IMS there are some steps to complete. First of all, it is needed to have the UEs properly registered in the HSS of LTEIMS of Amarisoft as described in the paragraph 3.3. If commercial SIM cards are planned to be used, it is important to keep in mind that:

- Secret values of the SIM must be known in order to subscribe the SIM into the ue_db-ims.cfg file.
- In case the secret values are unknown, it is possible to skip the authentication and the security operations in the attachment procedure by adding some commands depending on whether the EEA0/EIA0 encryption algorithms are supported by the device under test or not. If EEA0/EIA0 are supported, then two commands are needed: authentication_mode:"skip" in the mme.cfg file and skip_smc_proc:true in the enb.cfg file. If EEA0/EIA0 are not supported, then three commands are necessary: both authentication_mode:"skip" and skip_smc_proc:true in the mme.cfg file, and skip_smc_proc:true in the enb.cfg file. The encryption algorithms supported by the UE can be found in the attach request message, under UE network capability. See chapter 4.2.
- Disabling the authentication and security procedures is not allowed by the 3GPP specifications, hence even when writing the command to skip these procedures, a commercial UE will reject this command. The device under test, therefore, needs to be in a specific test mode. This modality is the only way for disabling the non-3GPP compliant behavior, but it is designed only by the device manufacturer.

Having listed all of the factors to take into account, for the sake of simplicity, in this thesis Amarisoft SIM cards have been used. The configuration of the database of the UEs will be discussed in the chapter 5.2 and it is showed in the Appendix under ue_db-ims.cfg title.

At the UE side, it is necessary to have both the internet APN and the IMS one. The Amari Classic Callbox has the following APNs are described in the following way:

- Name: "Internet".
- APN: "internet".
- APN type: "internet, default".

and

- Name: "IMS".
- APN: "ims".
- APN type: "ims".

In order to check that the device is correctly registered in the LTEIMS server, some useful commands from the screens can be used:

- (enb) cell phy

```
    [enb1a2d0] PLMN=00101 eNB_ID=0x1a2d0
--------Global--   -------DL----------  ---------UL--------
Cell  RAT BAND BW ARFCN ANT NL SCS QAM ARCFN ANT NL SCS QAM
0x001 LTE 7    5  3350  1   1  15  256 21350 1   1  15  64
0x002 LTE 3    5  1575  1   1  15  256 19575 1   1  15  64
```

Here, for the purpose of this thesis, it is important to check that the Radio Access Technology is LTE.

- (enb) ue

```
RAN_UE_ID      CN_UE_ID  Cell    RNTI
       579             409        0x001     0x027f
       577             407        0x001     0x027d
```

Only the UEs that have been properly registered in the eNB are given a RAN_UE_ID.

- (mme) ue

```
SUPI                    IMEISV  CN  REG IP_ADDR
001010123456789 8...304   EPC   Y    192.168.3.6 192.168.4.6
001010123456789 8...504   EPC   Y    192.168.3.2 192.168.4.2
```

Only the DUTs that have been properly registered (Y) are given an IP_ADDR.

- (ims) users

```
IMPI:    001010123456789@ims.mnc001.mcc001.3gppnetwork.org
  SIP Binding:
URI:sip:941da26a-a39a-4862-a134-
0d62b7db00a0@[2001:468:3000:1:1432:1fff:feb2:d872]:5060
    IMEI:    866929050364850
    Prio:    1.0
    Expires: 3472s
    Options: sms video volte
    IMPU:    sip:001010123456789 tel:0601
  MME:     127.0.0.2:10042 id=12
  SIP Binding:
URI:sip:605ff4de-9da7-4885-a15a-
10837a2936af@[2001:468:3000:2:78bc:4aff:fe79:255e]:5060
    IMEI:    866929050363730
    Prio:    1.0
```

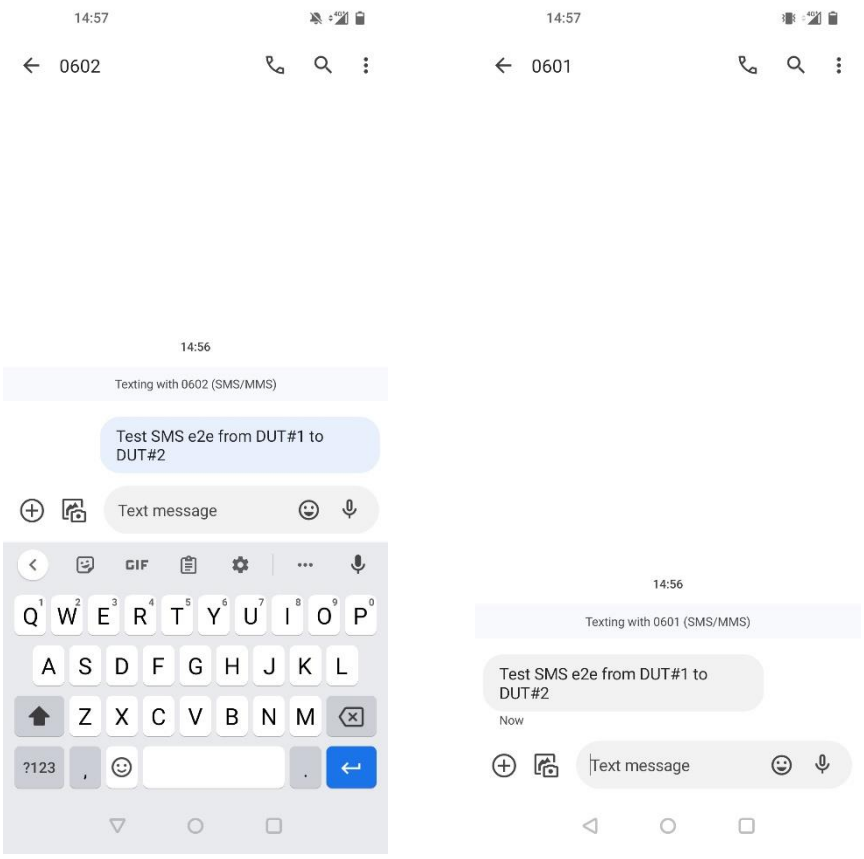
```
   Expires: 3441s
   Options: sms video volte
   IMPU:    sip:001010123456789 tel:0602
MME:     127.0.0.2:10042 id=11
```

Here, among the outputs there are some relevant ones: the “SIP Binding” message, the allowed options for each DUT (sms, video, volte), the IMPU’s tel number. In particular, the IMPU’s tel is the number that will be used to send SMS.

There are three modalities to test the sending of SMS via IMS:

- Mobile Oriented (MO), when a DUT sends an SMS via IMS to itself.
- Mobile Terminated (MT), when the SMS is sent by the IMS to the UE.
- End-to-end SMS test, when DUT#1 sends a SMS to DUT#2 via IMS.

In the end-to-end test, it is sufficient to send the desired text to the tel number of the desired IMPU.

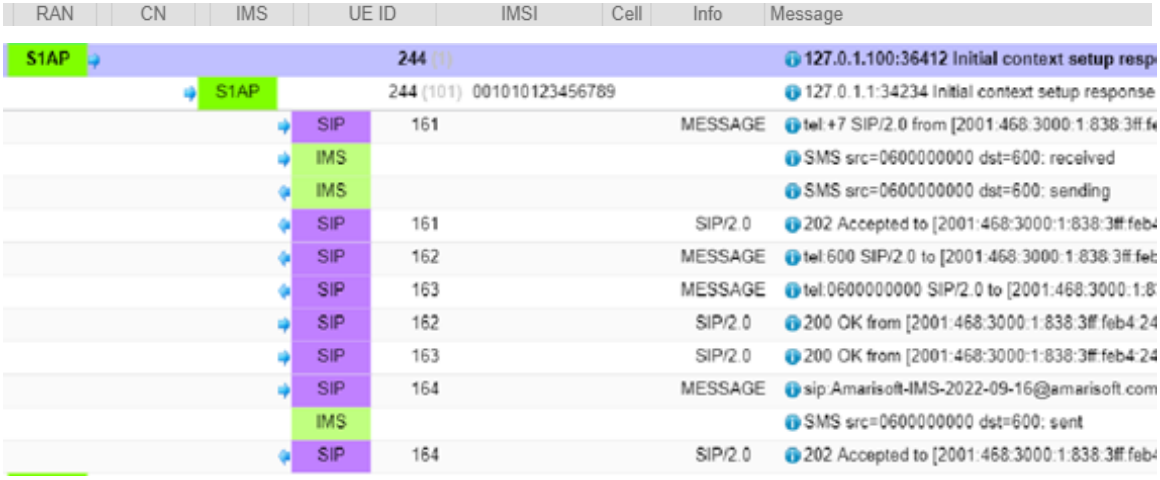


For example, in the pictures it is observable the sending of the message “Test SMS e2e from DUT#1 to DUT#2”:

- At the left side the DUT#1 sends the SMS to DUT#2 (tel number 0602).

- At the right side the DUT#2 receives the SMS from the DUT#1 (tel number 0601).

As a result, on Amarisoft web interface, in addition to the SIP messages, three IMS messages are generated: received, sending and sent. The generation of three IMS messages happens both in Mobile Oriented and in the end-to-end case.



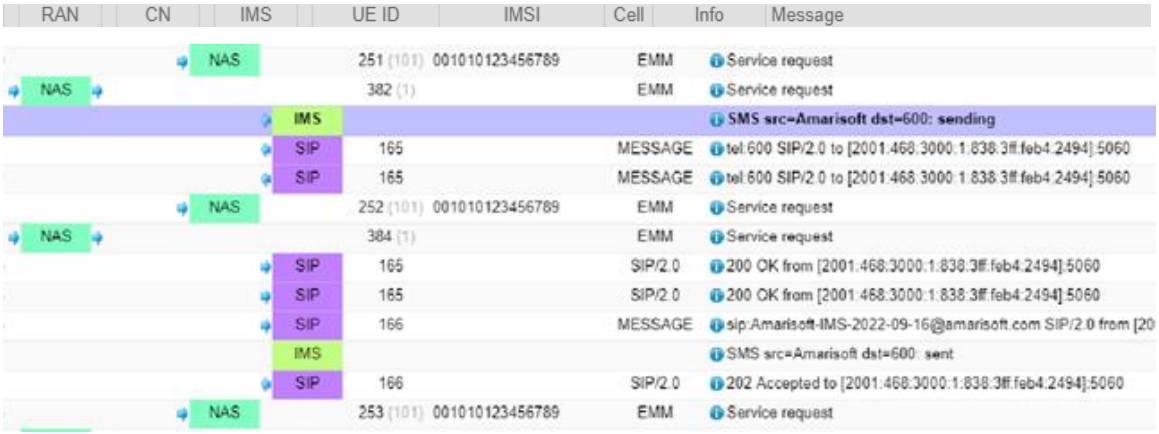
In the Mobile Terminated case, the message to the desired UE is sent through the screen of IMS.

For instance, the screenshot above is the result of the execution of the command

```
(ims) sms 0601 "Test SMS MT from IMS to DUT#1"
```

from the screen of LTEIMS.

As a consequence, on Amarisoft web interface, other than the SIP messages, two IMS messages are generated as well: sending and sent. This is due to the fact that LTEIMS is generating the SMS so it does not need to receive the message to send.



4.4 - VoLTE

The voice services delivered thanks to the IP technology substituted the circuit-switched technology that had a simpler architecture but it needed dedicated resources for each voice call, which meant that only the termination of the call could release the resources.

As mentioned previously, Amarisoft LTEIMS provides the framework for multimedia services that are IP based in a mobile network. The concept of IMS was introduced by 3GPP Technical Specifications and uses Internet Engineering Task Force (IETF) protocols, in particular, the SIP protocol.

The introduction of the IP Multimedia Subsystem allows to turn to calls using the IP technology. The signaling protocol, used by IMS, has diverse tasks such as: managing and controlling voice calls, video calls and instant messaging.

The VoLTE solution is based on IMS/MMTel services. The IMS core does not provide telephony, the equivalent of IMS for telephony is the global 3GPP standard Multimedia Telephony (MMTel) that is based on IMS and offers mobile real-time multimedia communication such as voice calls.

Devices that intend to start a VoLTE session have to support multiple default EPS bearers given that this type of session requires multiple EPS bearers at the same time.

The default EPS bearer is used for SIP signaling (QCI=5), while the dedicated EPS bearer is utilized for voice IP packets (QCI=1). In addition to these two bearers, another dedicated EPS bearer might be needed, for instance for a video call it is needed a bearer with QCI=2.

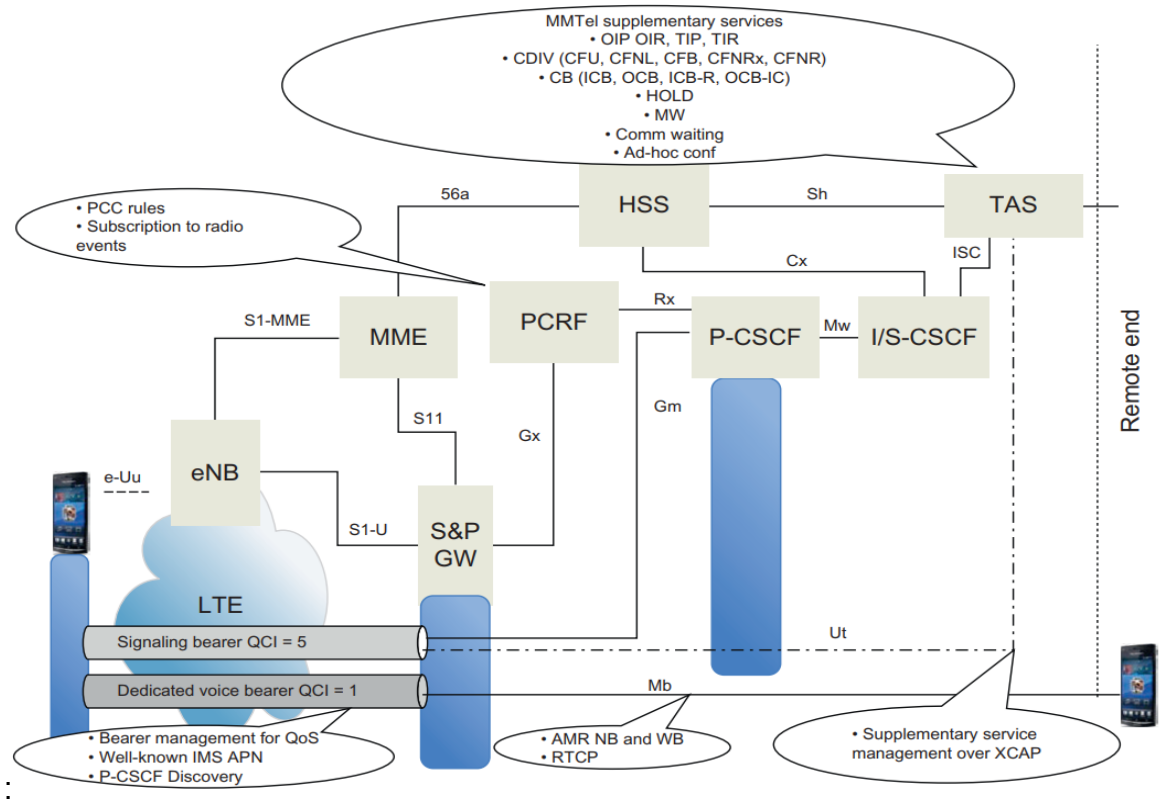


Figure: 4.4.1: VoLTE Architecture.

In order to be able to perform VoLTE calls using the Amari Callbox, it is required to configurate both the UE and the network. All the scenarios implemented for this thesis support VoLTE calls and it is possible to find the code of the different files in the Appendix.

At the UE side, the steps to follow under Settings/ Wi-Fi&Network/ SIM&Network/ SIM1 are:

- turn Mobile data ON;
- turn VoLTE ON;
- turn Roaming ON;
- create “Internet” and “IMS” APNs, for the creation of these APNs it is possible to refer to the chapter 4.3;
- select the “Internet” APN;

- reboot of DUT.

For what regards the network configuration, the files ims.cfg, mme.cfg, and others need to be properly configured. The ims.cfg file comprises many parameters like the SIP addresses array called “sip_addr” and the echo numbers “666” and “+666” which are the numbers used by the UE when it wants to call itself.

It is important to execute the command `service lte restart` after modifying the configuration files and it is advised to check whether the UE is properly registered in the IMS, namely, whether it has been assigned an IP address or not:

- (mme) ue

```
SUPI                    IMEISV  CN  REG IP_ADDR
001010123456789 8...304   EPC   Y    192.168.3.6 192.168.4.6
001010123456789 8...504   EPC   Y    192.168.3.2 192.168.4.2
```

In this output, only the devices that have been properly registered (Y) are given an IP_ADDR.

- (ims) users

```
IMPI:    001010123456789@ims.mnc001.mcc001.3gppnetwork.org
  SIP Binding:
URI:sip:941da26a-a39a-4862-a134-
0d62b7db00a0@[2001:468:3000:1:1432:1fff:feb2:d872]:5060
    IMEI:    866929050364850
    Prio:    1.0
    Expires: 3472s
    Options: sms video volte
    IMPU:    sip:001010123456789 tel:0601
  MME:     127.0.0.2:10042 id=12
  SIP Binding:
URI:sip:605ff4de-9da7-4885-a15a-
10837a2936af@[2001:468:3000:2:78bc:4aff:fe79:255e]:5060
    IMEI:    866929050363730
    Prio:    1.0
    Expires: 3441s
    Options: sms video volte
    IMPU:    sip:001010123456789 tel:0602
  MME:     127.0.0.2:10042 id=11
```

Here, among the outputs there are some relevant ones: the “SIP Binding” message, the allowed options for each DUT (sms, video, volte), the IMPU’s tel

number. In particular, the IMPU's tel is the number that will be used to initiate VoLTE calls in the end-to-end case.

After having configured the files of the different components of Amari Callbox, it is possible to test VoLTE calls in three modalities:

- Mobile Oriented (MO), in this case the "echo" test is performed by digiting the echo number from the device under test.
- Mobile Terminated (MT), the call is initiated by the IMS server. It is sufficient to digit `mt_call <tel IMPU>` from the screen of IMS.
- End-to-end call test, one DUT#1 starts the VoLTE call towards another DUT#2 by dialing the desired tel IMPU. This is the analyzed test of this thesis.

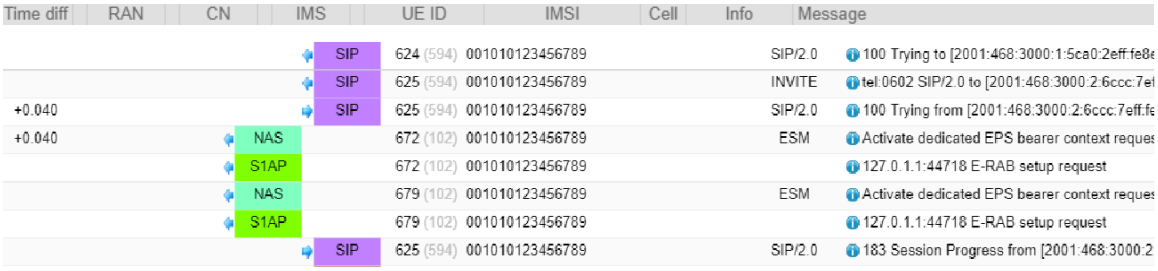
In the following analysis, the devices will be referred as DUT#1 and DUT#2, whereas the IMS is the same for both of the devices, it will be, therefore, referred as IMS without any index.

When the DUT#1 dials the DUT#2, the VoLTE call flow starts with an INVITE SIP message, that includes the first Session Description Protocol (SDP) offer, from the DUT#1 to the IMS. If needed, there can be a second INVITE message with a second offer until the desired UE is authenticated. The SDP is the protocol that describes a multimedia session in order to help users to join a session and it is composed by a sequence of lower-case letters which define the protocol version, the identification of the creator of the session, the session name, the phone number, bandwidth information, and others.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+10.541			SIP	623 (594)	001010123456789		INVITE	tel:0602;phone-context=ims.mnc001.mcc0
			CX					127.0.1.100:3868 Multimedia-Auth-Request
			CX					127.0.1.100:3868 Multimedia-Auth-Answer
			SIP	623 (594)	001010123456789		SIP/2.0	401 Unauthorized to [2001:468:3000:1:5ca0::
+0.039			SIP	623 (594)	001010123456789		ACK	tel:0602;phone-context=ims.mnc001.mcc001
+0.041			SIP	624 (594)	001010123456789		INVITE	tel:0602;phone-context=ims.mnc001.mcc001
			IMS	624 (594)	001010123456789			001010123456789 authenticated

Once the desired UE is authenticated, the 100 Trying message is sent to the IMS from DUT#2, in order to stop the INVITE messages. The DUT#2, after the activation of the dedicated EPS bearers, sends to the IMS the 183 Session Progress (SP) SIP message that contains the answer to the SDP sent by the DUT#1. This message of the DUT#2 gives information about the codec supported by it which gives information about the quality of the audio. For a good traditional telephony quality was sufficient a

codec of type Adaptive Multi-Rate Narrowband (AMR-NB), while in order to have a good VoLTE call quality it is necessary to have a codec of type Adaptive Multi-Rate Wideband (AMR-WB).



Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
			SIP	624 (594)	001010123456789		SIP/2.0	100 Trying to [2001:468:3000:1:5ca0:2eff:fe8
			SIP	625 (594)	001010123456789		INVITE	tel:0602 SIP/2.0 to [2001:468:3000:2:6ccc:7ef
+0.040			SIP	625 (594)	001010123456789		SIP/2.0	100 Trying from [2001:468:3000:2:6ccc:7eff:fe
+0.040		NAS		672 (102)	001010123456789		ESM	Activate dedicated EPS bearer context reques
		S1AP		672 (102)	001010123456789			127.0.1.1:44718 E-RAB setup request
		NAS		679 (102)	001010123456789		ESM	Activate dedicated EPS bearer context reques
		S1AP		679 (102)	001010123456789			127.0.1.1:44718 E-RAB setup request
			SIP	625 (594)	001010123456789		SIP/2.0	183 Session Progress from [2001:468:3000:2

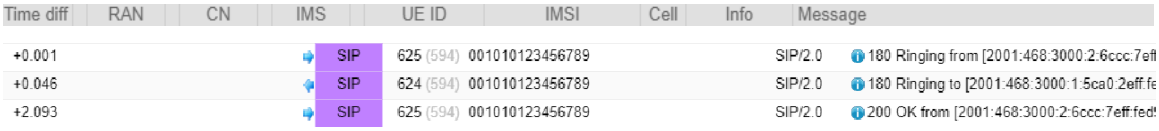
The calling device will then send the Provisional Response Acknowledgment (PRACK) SIP message to the IMS, as an answer to the 183 SP SIP message sent by DUT#2; this message will contain the voice codec that the DUT#1 has selected. Moving on, the called device sends a 200 OK SIP message where it accepts the selected voice codec chosen by the DUT#1.

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.013			SIP	624 (594)	001010123456789		PRACK	sip:001010123456789@[2001:468:3000:1::]:5
			SIP	624 (594)	001010123456789		SIP/2.0	200 OK to [2001:468:3000:1:5ca0:2eff:fe8e:c

The DUT#1, therefore, reserves its resources and it sends an UPDATE SIP message. The DUT#2 answers with a 200 OK SIP message confirming its reserved resources for the voice call.

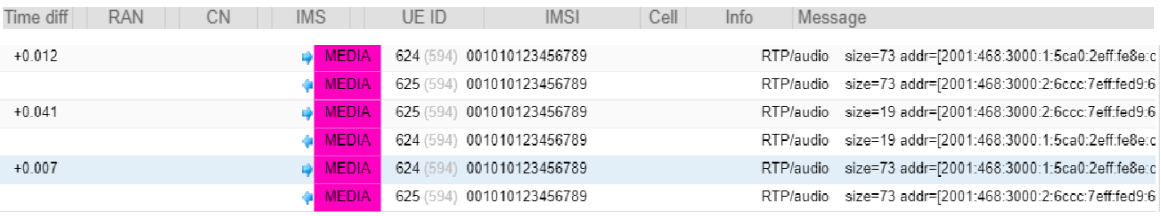
Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.249			SIP	625 (594)	001010123456789		UPDATE	tel:0602 SIP/2.0 to [2001:468:3000:2:6ccc:7e
+0.102			SIP	625 (594)	001010123456789		SIP/2.0	200 OK from [2001:468:3000:2:6ccc:7eff:fed9

After the creation of the EPS bearers and the reservation of the resources, the calling device can finally alert the called device that it is receiving a call by starting ringing. When the called device will answer, it will send a 200 OK SIP message to the DUT#1.

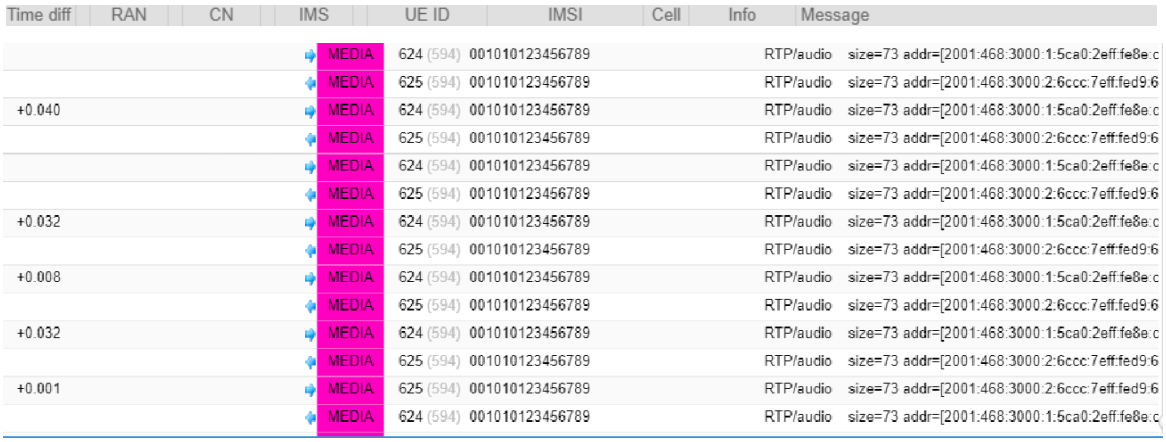


Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.001			SIP	625 (594)	001010123456789		SIP/2.0	180 Ringing from [2001:468:3000:2:6ccc:7eff
+0.046			SIP	624 (594)	001010123456789		SIP/2.0	180 Ringing to [2001:468:3000:1:5ca0:2eff:fe
+2.093			SIP	625 (594)	001010123456789		SIP/2.0	200 OK from [2001:468:3000:2:6ccc:7eff:fed9

Once the call starts, the exchange of Real Time Protocol (RTP) messages starts.

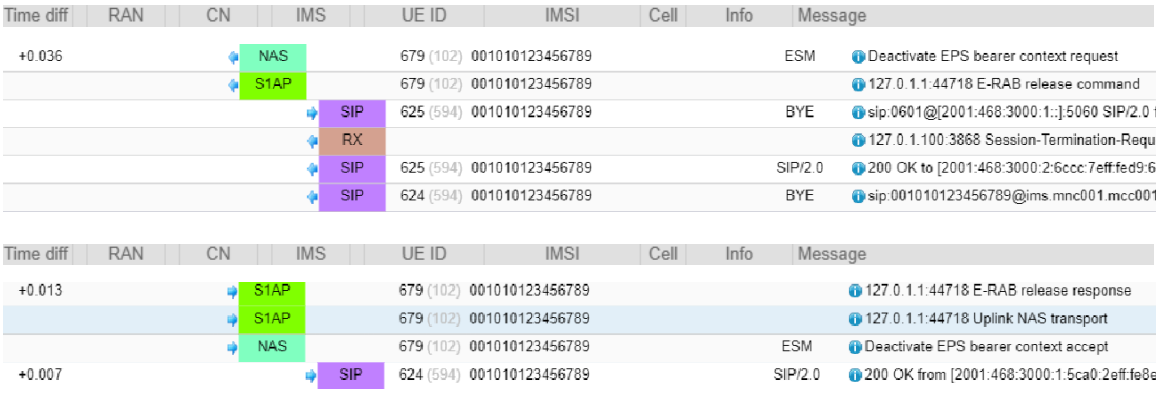


Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.012			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
+0.041			MEDIA	625 (594)	001010123456789		RTP/audio	size=19 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
			MEDIA	624 (594)	001010123456789		RTP/audio	size=19 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
+0.007			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6



Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
+0.040			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
+0.032			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
+0.008			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
+0.032			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c
			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
+0.001			MEDIA	625 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:2:6ccc:7eff:fed9:6
			MEDIA	624 (594)	001010123456789		RTP/audio	size=73 addr=[2001:468:3000:1:5ca0:2eff:fe8e:c

Supposing that the device that ends the call is the DUT#1, first the deactivation of the EPS bearers starts, then the DUT#1 sends the BYE SIP message. The DUT#2 replies with a 200 OK SIP message and a BYE SIP message. After the deactivation of the EPS bearers is complete, DUT#1 answers with a 200 OK SIP message which concludes the VoLTE call.



Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.036		NAS		679 (102)	001010123456789		ESM	Deactivate EPS bearer context request
		S1AP		679 (102)	001010123456789			127.0.1.1:44718 E-RAB release command
			SIP	625 (594)	001010123456789		BYE	sip:0601@[2001:468:3000:1::]:5060 SIP/2.0
			RX					127.0.1.100:3868 Session-Termination-Requ
			SIP	625 (594)	001010123456789		SIP/2.0	200 OK to [2001:468:3000:2:6ccc:7eff:fed9:6
			SIP	624 (594)	001010123456789		BYE	sip:001010123456789@ims.mnc001.mcc001

Time diff	RAN	CN	IMS	UE ID	IMSI	Cell	Info	Message
+0.013		S1AP		679 (102)	001010123456789			127.0.1.1:44718 E-RAB release response
		S1AP		679 (102)	001010123456789			127.0.1.1:44718 Uplink NAS transport
		NAS		679 (102)	001010123456789		ESM	Deactivate EPS bearer context accept
+0.007			SIP	624 (594)	001010123456789		SIP/2.0	200 OK from [2001:468:3000:1:5ca0:2eff:fe8e

4.5 - Handover

Amarisoft allows to test handover in different ways. Handover can be tested either by using a commercial UE or by using a UE simulator that can be bought from Amarisoft. In this thesis handover has been tested using a commercial user equipment.

Another way of distinguishing handover tests is based on how many eNBs are utilize. The handover can be tested either intra eNodeB or inter eNodeB. In the latter case two different hosts are needed because each eNodeB must run on a different pc and, therefore, an additional license is needed because each eNB requires its own license which of course has a further cost. For these reasons, in this thesis only intra eNodeB Handover has been tested. For this test one OnePlus 9 Pro device has been utilized,

the user equipment is connected to the LTEENB that is connected to the LTEMME via the S1 interface.

When testing inter eNB handover namely, when testing handover between two cells running on the same eNB, depending on the width of the band used by both cells, one or two SDR cards might be required.

In the data sheet of Amari Callbox Classic there are the specifications of the Peripheral Component Interconnect Express (PCIe) SDR, for example it is possible to find the RF bandwidth of each SDR card which goes from 200kHz to 56MHz. A maximum bandwidth of 56 MHz means that if both cells fit in this bandwidth, then handover can be carried out with one SDR card only, otherwise two SDR cards are required in order to carry out handover.

In the following example, the two cells have bandwidth small enough so that, when using the same SDR card, there is no overlap between the two cells neither in downlink nor in uplink.

Example of 2 cells fitting in one SDR cards :

- Cell1 (Band7) , dl_earfcn: 3350, 2680,0Mhz DL-2560,0Mhz UL freq, 5Mhz Bandwidth
- Cell2 (Band7) , dl_earfcn: 3299, 2674.9Mhz DL-2554.9Mhz UL freq, 5Mhz Bandwidth

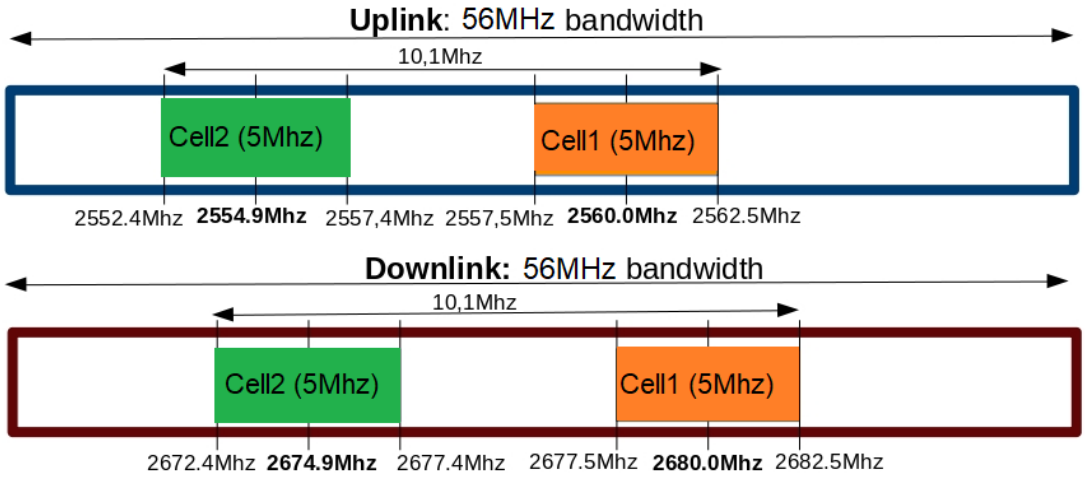


Figure 4.5.1: Example two cells in the same SDR.

If the two cells would have had larger bandwidth, for instance 20 MHz, they would have overlapped which would have meant that in order to avoid interference two SDR cards would be been needed.

There are others constraints in addition to the bandwidth one:

- The difference of the center frequencies of each cell has to be a multiple of 300 kHz i.e., the4 difference of their EARFCN has to be a multiple of 3.
- The difference between the center frequency of each cell and the average of center frequencies needs to be a multiple of 15 kHz.
- The Physical Random Access Channel (PRACH) of the cells must have the same duration and they must be transmitted in the same subframes.
- In the current version of Amarisoft, multiple cells can be set at the same frequency provided their physical cell identity (n_id_cell) and PRACH root sequence index are different in order to get a minimum inter-cell interference. Nevertheless, it is important to keep in mind that resources cannot be reserved by the cells, thus if different cells transmit at the same time and in the same resource blocks a performance degradation would happen. For this reason, in the current version of Amarisoft, it is better to use cells at different frequencies.
- The number of cells that can be configured in a frequency bandwidth depends on the total bandwidth of the LTE band and it depends also on the bandwidth of each cell plus the offsets.

Before start testing, it is needed to define each cell object inside the cell list, called cell_list, in the enb.cfg file. It is also relevant to set, for each cell, information about its neighbour cells by using the list labelled ncell_list.

The configuration of the list “cell_list” changes depending on how many SDR cards are used. In the following pages there are two examples:

- Example A: two cells in the same SDR card.
- Example B: two cells in two different SDR cards.

In both examples, each neighbour has the n_id_cell parameter which indicates the physical cell identity of the neighbour cell, followed by the dl_earfcn, used by the neighbour, and others.

Example A:

```
cell_list:                          [
  {
    dl_earfcn: 3350,   /* DL center
frequency: 2680 MHz (Band 7) */

    n_id_cell:                  1,
    cell_id:                 0x01,
```

Example B:

```
cell_list:                          [
  {
    rf_port: 0, /* means that cell
0x01 will use dev0=/dev/sdr0 */
    dl_earfcn: 3350,   /* DL center
frequency: 2680 MHz (Band 7) */
    n_id_cell:                  1,
```

```
    tac:                        0x0001,
    root_sequence_index:   204,   /*
PRACH   root   sequence   index   */

    /* Neighbour cell list (used for
handover)                           */
    ncell_list:                     [
      {  n_id_cell:  2,  dl_earfcn:
3299, cell_id: 0x1a2e002, tac: 1 },
    ],
  },
  {
    dl_earfcn:  3299,  /*  DL  center
frequency:  2674.9  MHz  (Band  7)  */

    n_id_cell:                      2,
    cell_id:                     0x02,
    tac:                        0x0001,
    root_sequence_index:   86,   /*
PRACH   root   sequence   index   */

    /* Neighbour cell list (used for
handover)                           */
    ncell_list:                     [
      {  n_id_cell:  1,  dl_earfcn:
3350, cell_id: 0x1a2e001, tac: 1 },
    ],
  },
  ], /* cell_list */
```

```
    cell_id:                     0x01,
    tac:                        0x0001,
    root_sequence_index:   204,   /*
PRACH   root   sequence   index   */
    /* Neighbour cell list (used for
handover)                           */
    ncell_list:                     [
      {  n_id_cell:  2,  dl_earfcn:
3299, cell_id: 0x1a2e002, tac: 1 },
    ],
  },
  {
    rf_port: 1,  /* means that cell
0x02  will  use  dev1=/dev/sdr1  */
    dl_earfcn:  3299,  /*  DL  center
frequency:  2674.9  MHz  (Band  7)  */
    n_id_cell:                      2,
    cell_id:                     0x02,
    tac:                        0x0001,
    root_sequence_index:   86,   /*
PRACH   root   sequence   index   */

    /* Neighbour cell list (used for
handover)                           */
    ncell_list:                     [
      {  n_id_cell:  1,  dl_earfcn:
3350, cell_id: 0x1a2e001, tac: 1 },
    ], /* cell_list */
```

As can be noticed, in the example where two SDR cards are utilized, each cell is defined with its own rf_port. The example B, therefore, needs a further configuration in the ./rf_driver/config.cfg file:

```
rf_driver: {
    name: "sdr",

    /* list of devices. 'dev0' is always the master. */
    args: "dev0=/dev/sdr0,dev1=/dev/sdr1",
```

If three SDR cards are used, the list called cell_list includes the definition of the cell for rf_port: 2 and in the rf_driver/config.cfg file the args parameter changes to:

```
args: "dev0=/dev/sdr0,dev1=/dev/sdr1,dev2=/dev/sdr2"
```

When using more than one SDR card it is advised to check that:

(i) all SDR cards use the same FW;

(ii) all SDR cards use the version of FW compatible with the SW of the eNB;

(iii) the speed of the PCIe link is higher than the threshold.

In the interest of monitoring the FW used by the SDR cards, it is possible to use the command `./sdr_util version` from the trx_sdr directory which can be found

inside the root directory. Among the lines of the output, it is possible to notice the type of SDR used by the callbox and the software version used by the three SDR cards. It is important to stop the LTE service before executing the command for the SDR version.

```
=== Device /dev/sdr0 ===
Board ID:          0x4b01 (SDR50)
Board master:      0x0
Board revision:    0x0
FPGA revision:     2022-07-13 08:15:02
FPGA status:       operational
Software version: 2022-09-13
DMA:               1 ch, 64 bits, SMem index: On
DNA:               [0x20447761543374932]
Serial             ''
PCIe bus:          bus=0x01 FPGA PCI gen2 x1 (4.0Gb/s)    OK

=== Device /dev/sdr1 ===
Board ID:          0x4b01 (SDR50)
Board master:      0x0
Board revision:    0x0
FPGA revision:     2022-07-13 08:15:02
FPGA status:       operational
Software version: 2022-09-13
DMA:               1 ch, 64 bits, SMem index: On
DNA:               [0x20447761543374932]
Serial             ''
PCIe bus:          bus=0x04 FPGA PCI gen2 x1 (4.0Gb/s)    OK

=== Device /dev/sdr2 ===
Board ID:          0x4b01 (SDR50)
Board master:      0x0
Board revision:    0x0
FPGA revision:     2022-07-13 08:15:02
FPGA status:       operational
Software version: 2022-09-13
DMA:               1 ch, 64 bits, SMem index: On
DNA:               [0x24933768984701012]
Serial             ''
PCIe bus:          bus=0x03 FPGA PCI gen2 x1 (4.0Gb/s)    OK
```

In order to check that the firmware version of the SDR cards is compatible with the eNodeB software, it is possible to update the SDR devices via the command `./sdr_util upgrade` from the trx_sdr directory under the root directory. This command will trigger a firmware update only in case the FW of the SDR cards is not compatible with the eNB software.

For what concerns the speed of the PCIe link, it is possible to check it through the command `./sdr_test dma_loopback_test n` where n stands for the number of SDR cards. In this thesis case study, given that the classic callbox uses SDR50 cards, the dma throughput should be greater-than-or-equal-to 3.4Gb/sec. For SDR100 cards this threshold increases to 11.3Gb/sec.

```
#0 sdr2 3.4 Gb/sec 25641.0 HFN/sec tx_underflows=0 load: W=6.0%
R=0.0% W+R=6.0% T=50.8
#0 sdr0 3.4 Gb/sec 25641.0 HFN/sec tx_underflows=0 load: W=6.0%
R=0.0% W+R=6.0% T=49.7
#0 sdr1 3.4 Gb/sec 25641.0 HFN/sec tx_underflows=0 load: W=6.0%
R=0.0% W+R=6.0% T=52.8
...
#9 sdr1 3.4 Gb/sec 25641.0 HFN/sec tx_underflows=0 load: W=6.3%
R=0.0% W+R=6.3% T=53.0
```

For handover one of the most important steps is the measurement report from the DUT before the handover takes place. For this reason, it is necessary to set meas_config_desc parameters, in the cell_default object that it is placed inside the enb.cfg file, in order to trigger the measurement reports at UE side when the level of the cells will fluctuate.

In all technologies measurement reports are generated by UE and the criteria for the measurement is determined by the RRC messages. From the point of view of the RRC messages the measurement process comprises:

- RRC message sent by eNB indicating the values to be measured.
- RRC message sent by UE carrying the measurement results.

The network will decide whether it will let the UE handover or not based on the measurement values from the device. 3GPP defines diverse sets of measurement report to be performed by the user equipment, these measurement reports are called “event”. The type of event the UE has to perform is specified in the RRC connection Reconfiguration message.

Amarisoft utilizes three types of events: A1, A2 and A3.

Event Type	Description
Event A1	Serving becomes better than threshold
Event A2	Serving becomes worse than threshold
Event A3	Neighbour becomes offset better than serving

In the cell_default object the measurement configuration parameters are set as follows:

```
/* measurement configuration */
meas_config_desc: {
a1_report_type: "rsrp",
a1_rsrp: -70,
a1_hysteresis: 0,
a1_time_to_trigger: 640,
a2_report_type: "rsrp",
a2_rsrp: -80,
a2_hysteresis: 0,
a2_time_to_trigger: 640,
eutra_handover: {
a3_report_type: "rsrp",
a3_offset: 6,
hysteresis: 0,
time_to_trigger: 480
}
},
/* measurement gap configuration */
meas_gap_config: "gp0",
/* if true, initiate a handover when a suitable measurement
report
is received */
ho_from_meas: true,
```

In this configuration, it is visible that the value measured is the RSRP, each event has its own threshold and its hysteresis.

The hysteresis is a small fluctuation range that can be introduced in order not to get unnecessarily frequent measurement reports. If hysteresis is set equal to zero, the measured values will cause a lot of measurement reports; while if it is set not equal to zero, the measured values will cause less measurement reports.

Another important parameter is the measurement GAP, this parameter creates a small gap where neither transmission nor reception happen. This gap is introduced in order to allow the UE to switch to the target cell to perform the measurement and then switch back to the current cell.

In amarisoft, the default value is gp0, which refers to an integer between 0ms and 39ms.

After all the modifications in the enb.cfg file, it is crucial to execute the `service lte restart` command in order to update the eNB block of the Amari Callbox Classic.

Handover can be triggered by:

- UE measurement report.
- The eNodeB (Blind Handover case).

In both cases, in order to keep the user equipment in RRC connected state, it is advised to perform handover while running uplink or downlink transfer using either iperf or the LTE simulation server provided by Amarisoft.

For the sake of the performing handover based on measurement report from the UE, it is necessary to decrease the gain at the eNodeB side. The command to execute from the screen of the eNB is:

```
cell_gain <cell_id> <gain>
```

where the cell_id parameter is the identification of the cell to be abandoned. An example can be: `cell_gain 1 -20`. The cell_gain parameter is the gain of the cell in downlink in dB, its range goes from –200dB to 0dB, and its default value is zero.

In the interest of performing blind handover, the command to execute from the screen of the eNodeB is:

```
handover <eNB_UE_ID> <pci> <dl_earfcn> <type>
```

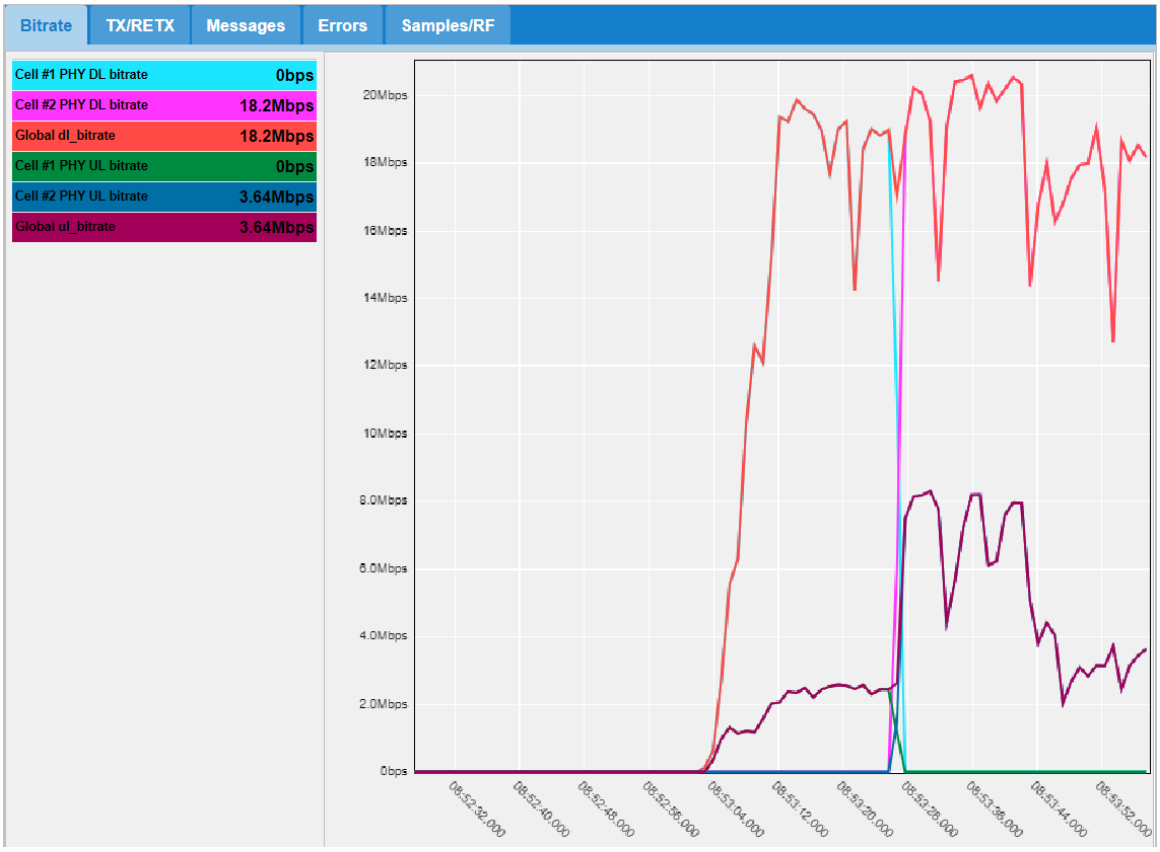
where:

- The `eNB_UE_ID` can be obtained from the eNB screen by typing the `ue` command. It is important to get this value only after the iperf command has started.
- The `pci` parameter is the physical cell identity of the neighbor cell (the target cell).
- The `dl_earfcn` parameter is the downlink EARFCN of the neighbor cell (the target cell).
- The `type` parameter in this thesis case is always `intra`.

An example of handover command for the blind case is:

```
handover 160 2 1575 intra
```

this command gives as a result the following plot.


Bitrate
TX/RETX
Messages
Errors
Samples/RF
Cell #1 PHY DL bitrate 0bps
Cell #2 PHY DL bitrate 18.2Mbps
Global dl_bitrate 18.2Mbps
Cell #1 PHY UL bitrate 0bps
Cell #2 PHY UL bitrate 3.64Mbps
Global ul_bitrate 3.64Mbps
20Mbps
18Mbps
16Mbps
14Mbps
12Mbps
10Mbps
8.0Mbps
6.0Mbps
4.0Mbps
2.0Mbps
0bps
08:52:32.000
08:52:40.000
08:52:48.000
08:52:56.000
08:53:04.000
08:53:12.000
08:53:20.000
08:53:28.000
08:53:36.000
08:53:44.000
08:53:52.000

5 – Carrier Aggregation and Throughput

5.1 - Throughput theory

In mobile networks, LTE provides an increase of the capacity in data transmission thanks to the introduction of technologies such as Multi-Input Multiple-Output and Carrier Aggregation. The throughput can be measured in Downlink only, in Uplink only, or both in Downlink and in Uplink at the same time, and it can be measured having as transport protocol either UDP or TCP. From the literature, the characteristics of these transport protocols are known.

The TCP protocol attributes are:

- reliable data transfer;
- throughput sensible to the end-to-end delay;
- throughput sensible to packet losses;
- downlink throughput limited by its Uplink throughput;
- reception of the TCP ACK in UL/DL has a negative impact on the throughput in DL/UL.

The UDP protocol attributes are:

- not as reliable as TCP protocol;
- used in multimedia applications;
- used in real-time applications;
- throughput is not sensible to the end-to-end delay;
- the throughput in one direction is independent from what happens in the other direction.

It is well known that in order to achieve the maximum throughput there are relevant factors to take into account, for example:

- UE category.
- Configurations of application level.
- Network configuration: bandwidth, modulation scheme, number of resource blocks (RBs), number of carriers and number of layers.
- Number of users in one cell.

- Level of the signal: channel and fading conditions, link losses, noise from other devices, interference from other Base Stations or from other RANs.

For what concerns the UE category, the first thing to do is to check the RRC message “UE Capability Information” that has been exchanged between the DUT and the eNB in the attachment phase. This RRC message contains, as shown, the category of the device, the uplink category, and the downlink category:

```
...
ue-Category 4,
...
ue-CategoryDL-r12 13,
ue-CategoryUL-r12 5,
...
```

This data is useful to understand which are the theoretical maximums for the throughput both in uplink and in downlink. These maximum values for the theoretical throughput do not give any details about the MIMO or CA configuration.

The 3GPP organization provides tables in which it divides devices in different categories, and it assigns different theoretical maximum throughputs to each category. More accurately, the 3GPP Technical Specification 36.306 version 13.5.0 encompasses the tables listed in the following pages.

The following table exhibits the uplink physical layer parameters set by the ue-Category of the user equipment.

UE Category	Maximum number of UL-SCH transport block bits transmitted within a TTI	Maximum number of bits of an UL-SCH transport block transmitted within a TTI	Support for 64QAM in UL
Category 1	5160	5160	No
Category 2	25456	25456	No
Category 3	51024	51024	No
Category 4	51024	51024	No
Category 5	75376	75376	Yes
Category 6	51024	51024	No
Category 7	102048	51024	No
Category 8	1497760	149776	Yes
Category 9	51024	51024	No
Category 10	102048	51024	No
Category 11	51024	51024	No
Category 12	102048	51024	No

Figure 5.1.1: UL physical layer parameter values set by ue-Category.

The following table illustrates the downlink physical layer parameters set by the ue-Category of the device.

UE Category	Maximum number of DL-SCH transport block bits received within a TTI (Note 1)	Maximum number of bits of a DL-SCH transport block received within a TTI	Total number of soft channel bits	Maximum number of supported layers for spatial multiplexing in DL
Category 1	10296	10296	250368	1
Category 2	51024	51024	1237248	2
Category 3	102048	75376	1237248	2
Category 4	150752	75376	1827072	2
Category 5	299552	149776	3667200	4
Category 6	301504	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	3654144	2 or 4
Category 7	301504	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	3654144	2 or 4
Category 8	2998560	299856	35982720	8
Category 9	452256	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	5481216	2 or 4
Category 10	452256	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	5481216	2 or 4
Category 11	603008	149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	7308288	2 or 4
Category 12	603008	149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	7308288	2 or 4

Figure 5.1.2: DL physical layer parameter values set by ue-Category.

The device under test used in this thesis is a category 4 device. Given that the ue-Category is four, it is able to support up to 150Mbps in downlink and up to 51 Mbps in uplink. From Release 12, a device can specify its category in downlink and uplink separately.

In the UE capability message, it is possible to find:

`ue-CategoryDL-r12 12.`

The OnePlus 9 device, therefore, declares to be category 12 in Downlink with 603 Mbps in DL.

The following table, shown in two parts, lists the physical layer parameter values analogous to the ue-CategoryDL.

UE DL Category	Maximum number of DL-SCH transport block bits received within a TTI (Note 1)	Maximum number of bits of a DL-SCH transport block received within a TTI	Total number of soft channel bits	Maximum number of supported layers for spatial multiplexing in DL
DL Category M1	1000	1000	25344	1
DL Category 0 (Note 2)	1000	1000	25344	1
DL Category 6	301504	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	3654144	2 or 4
DL Category 7	301504	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	3654144	2 or 4
DL Category 9	452256	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	5481216	2 or 4
DL Category 10	452256	149776 (4 layers, 64QAM) 75376 (2 layers, 64QAM)	5481216	2 or 4
DL Category 11	603008	149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	7308288	2 or 4
DL Category 12	603008	149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	7308288	2 or 4
DL Category 13	391632	195816 (4 layers, 256QAM) 97896 (2 layers, 256QAM)	3654144	2 or 4
DL Category 14	3916560	391656 (8 layers, 256QAM)	47431680	8
DL Category 15	749856-798800 (Note 3)	149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	9744384	2 or 4
DL Category 16	978960 -1051360 (Note 3)	149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	12789504	2 or 4
DL Category 17	25065984	391656 (8 layers, 256QAM)	303562752	8

Figure 5.1.3: DL physical layer parameter values set by ue-CategoryDL - Part 1.

UE DL Category	Maximum number of DL-SCH transport block bits received within a TTI (Note 1)	Maximum number of bits of a DL-SCH transport block received within a TTI	Total number of soft channel bits	Maximum number of supported layers for spatial multiplexing in DL
DL Category 18	1174752-1206016 (Note 3)	[299856 (8 layers, 64QAM) 391656 (8 layers, 256QAM)] 149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	14616576	2 or 4 [or 8]
DL Category 19	1566336 -1658272 (Note 3)	[299856 (8 layers, 64QAM) 391656 (8 layers, 256QAM)] 149776 (4 layers, 64QAM) 195816 (4 layers, 256QAM) 75376 (2 layers, 64QAM) 97896 (2 layers, 256QAM)	19488768	2 or 4 [or 8]

Figure 5.1.4: DL physical layer parameter values set by ue-CategoryDL - Part 2.

An ue-CategoryDL equal to twelve requires the support of modulation 256 QAM in downlink in order to allow transport blocks to carry up to 97896 bits when using two layers.

With respect to the uplink, the user equipment declares to be category 13 with 150 Mbps: `ue-CategoryUL-r12 13`.

The next table specifies the physical layer parameter values that correspond to the ue-CategoryUL.

UE UL Category	Maximum number of UL-SCH transport block bits transmitted within a TTI	Maximum number of bits of an UL-SCH transport block transmitted within a TTI	Support for 64QAM in UL
UL Category M1	1000	1000	No
UL Category 0	1000	1000	No
UL Category 3	51024	51024	No
UL Category 5	75376	75376	Yes
UL Category 7	102048	51024	No
UL Category 8	1497760	149776	Yes
UL Category 13	150752	75376	Yes
UL Category 14	9585664	149776	Yes
UL Category 15	226128	75376	Yes

Figure 5.1.5: UL physical layer parameter values set by ue-CategoryUL.

The declared category equal to thirteen in UL requires the support of modulation 64-QAM in uplink in the interest of transport blocks able to carry up to 75376 bits.

It is important to keep in mind that the Transmission Time Intervals (TTIs) carry user data, overhead, headers, and others. For this reason, the real throughput values will be always less than the theoretical values.

In pursuance of high throughput, the radio channel should be perfect. The throughput is visibly affected by the presence of interference with other bands, a Non-Line-of-Sight, and other factors which affect the channel. In order to minimize the interference with other bands when using antennas for transmission and reception, it is possible to use shielded boxes to isolate the user equipment from interference.

The number of users in one cell depends on the capacity of the cell itself. In case of one single UE, the cell could allocate all the resources of the cell to that single device. In case of more than one UE, the resources are shared among all the devices; it is up to the scheduling algorithm in the eNB to manage the sharing of the resources.

It is well known to everyone the correlation between bandwidth and throughput: the higher the bandwidth, the higher the throughput. In Amarisoft the bandwidth is set by setting the `n_rb_dl` parameter in the eNB configuration file. The `n_rb_dl` parameter can be equal to 6, 15, 25, 50, 75, 100 which refer to bandwidth values equal to 1.4, 3, 5, 10, 15, 20 MHz, respectively. Of course, in order to get the maximum value of throughput the bandwidth should be set to 20 MHz.

LTE has introduced advanced technologies such as MIMO in order to achieve higher data rates. In the version of Amarisoft studied in this thesis MIMO is not supported in Uplink, MIMO is supported in downlink only. The transmission mode is configured with two parameters in the eNB configuration file: `transmission_mode` and `transmission_mode_opt`.

The `transmission_mode` parameter has a range between 1 and 6, its default value is one which refers to a single antenna port. When more than one DL antenna are utilized, `transmission_mode` can be set to values between 2 and 6; values 3 and 4 need a Rank Indicator (RI) reporting that can be set via the `m_ri` parameter. This parameter sets the periodicity of the CQI/PMI reports. For two layers spatial

multiplexing the transmission_mode is set to 3 or 4 and this means that the m_ri parameter should be set to a non-zero value.

In downlink, it is required the support of 256-QAM as Modulation Coding Scheme (MCS) for this reason in the eNB configuration file it is necessary to specify the dl_256qam parameter and set it to true, given that its default boolean value is false.

In uplink, it is required the support of 64-QAM as MCS; in the enb.cfg file, therefore, it is necessary to set the ul_64qam parameter to true. If this parameter is not specified in the eNB configuration file or if the UE does not support 64-QAM in uplink, then the eNB is able to allocate up to MCS 24. If ul_64qam is specified and the UE supports 64-QAM in uplink, then the MCS goes up to 28.

For what concerns the number of resource blocks, it is relevant to keep in mind that not all resource blocks are allocated to user data. In DL, some resource blocks could be allocated to SIBs and physical downlink channels. In UL, some resource blocks could be allocated to CQI reports and physical uplink channels.

Amarisoft allows to specify the number of Orthogonal Frequency-Division Multiplexing (OFDM) symbols for the Physical Downlink Control Channel (PDCCH) through the n_symb_cch parameter in the enb.cfg file. This parameter has a range between 0 and 4; if it is equal to zero it means that the number of OFDM symbols will be automatically adjust; this value should not be used in a cell where cross carrier PDCCH signaling is enabled. It is, therefore, needed to have cross_carrier_scheduling equal to false in case n_symb_cch is equal to zero. The more UEs are present in one cell, the more OFDM symbols are needed for the PDCCH. In real life, n_symb_cch is set to values greater than or equal to zero.

In uplink, the nRB-CQI parameter allows to control the number of RBs allocated to CQI. The n1PUCCH-AN parameter allows to manage the number of RBs allocated to PUCCH. Both nRB-CQI and n1PUCCH-AN are in the configuration of SIB2, in the sib2_3.asn file.

If the RBs allocated for control in UL decrease, there is a gain in terms of UL throughput. It is possible to decrease the RBs allocated in uplink by setting the values of the parameters cqi_pucch_n_rb and n1_pucch_sr_count to 1. GIven that the more UEs are present in one cell, the more RBs the cell needs for CQI and PUCCH,

it is important to keep in mind that these two parameters should be set to values greater than one.

Once all the parameters are set at their best, the throughput can be tested in different modalities. A common tool is iPerf which is used for network testing and it can create TCP and UDP data streams. The iPerf tool has a client/server functionality and it is able to measure the throughput either bidirectionally or unidirectionally.

It is possible to test:

- DL throughput only
- Ul throughput only
- DL and UL simultaneously

The downlink test in MIMO of a device of category 4 in UDP involves:

- At UE side iperf in server mode: `iperf -s -u -i 1`
- At CN side iperf in client mode, getting the IP address of the UE from MME monitor: `iperf -c <UE IP address> -u -b 150M -i 1 -t 100`

For a TCP connection, it is sufficient to run the two commands, both at UE side and at core network side, by removing the red parts.

The uplink test in MIMO of a device of category 4 in UDP encompasses:

- At CN side iperf in server mode: `iperf -s -u -i 1`
- At UE side iperf in client mode, getting the IP address of the PDN GW from the mme.cfg file: `iperf -c <PGW_IPaddr> -u -b 50M -i 1 -t 100`

The PDN gateway can be not set for PDNs, in this case it will be used the first IP of the subnet in IPv4 as PGW. As for the downlink test, for a TCP connection, it is sufficient to run the two commands by removing the red parts.

The simultaneous uplink and downlink test expects either two iperf commands as the ones for the DL/UL test only, or the usage of the `-d` option. A TCP test comprises:

- At UE side iperf in server mode: `iperf -s -i 1`
- At CN side iperf in client mode, getting the IP address of the UE from the MME monitor: `iperf -c <UE IP addr> -d -i 1 -t 100`

After running the iperf commands in order to be able to see the bitrate in DL and in UL it is necessary to run the `t` command from the screen of the eNB. Good values of DL throughput are obtained when the CQI is equal to 15 and RI is equal to 2 in case of a MIMO system.

An interesting tool of Amarisoft is the LTE SimServer which, for the sake of throughput test, generates UDP packets. With LTESimServer is easier, with respect to iperf, to get throughput values closer to the maximum ones as long as the PHY throughput is high enough.

For accessing the simulation server, it is necessary to run `./ltesim_server` from the mme directory. Once, the LTE IP Simulation server is ON, it is possible to generate IP throughput through the following command:

```
cbr_send <UE IP> <data rate>M <test duration in seconds>
```

where:

- the IP of the UE can be obtained from the LTEMME screen;
- the data rate value is followed by a capitol M and it is set a little higher than the theoretical maximum throughput the UE can achieve;
- the test duration is in seconds.

An example of command is:

```
cbr_send 192.168.3.2 150M 600
```

this command sends UDP packets at constant bitrate in uplink for up to 600 seconds.

After running the cbr command, in the output of the t command, there are some crucial parameters to keep under observation such as the MCS value that has to be close to the maximum values: 28 in case of 64-QAM and 27 in case of 256-QAM.

If the CQI is too low or the retransmissions are too many, the MCS value will not reach the maximum value which means that the throughput will not reach high values. In order to keep the retx parameter low, the CQI high and the desired ri value, it is advised to check the radio link quality, i.e., the distance from the antennas, the antenna direction, the cable connection, and others.

5.2 - Carrier Aggregation theory

Carrier aggregation aggregates the traffic at MAC layer. In the interest of the aggregation a precise synchronization between the cells is vital. The steps for carrier aggregation to happen are three:

1. Measurement performance for all the relevant cells.
2. RRC connection reconfiguration to add Secondary Component Carriers (SCCs).
3. MAC Control Element (CE) activating SCCs.

Although the first step is always performed in real mobile networks, Amarisoft allows to choose whether to perform or not the first step, given that for the 3GPP organization the first step is always optional.

It is known that Carrier Aggregation can happen in three modalities:

(i) Intra-band and contiguous.
(ii) Intra-band and non-contiguous.
(iii) Inter-band and non-contiguous.

In contemplation of the writing of this thesis, three different tests of inter band CA have been performed:

- CA x2CCs without measurement.
- CA x2CCs with measurement report.
- CA x3CCs in DL and in UL without measurement.

In these tests each band requires its own PCIe SDR card. A meaningful aspect to take in consideration is that not all commercial phones support CA with three CCs in uplink. The UE used in this thesis supports maximum two CCs for CA in uplink. Moreover, only few combinations of bands allow to have CA x2. It is, therefore, crucial the choice of the bands for the cells in order to achieve high throughput due to carrier aggregation.

The CA test can involve or not the measurement report, if it is desired, one object and a parameter value are particularly relevant. The optional enumeration is `meas_gap_config` which is by default `none`. In case of carrier aggregation, the configuration of the measurement gap to `gp0` is a must since CA is always an interfrequency operation.

```
/* measurement gap configuration */
meas_gap_config: "gp0",
```

An example of measurement report object is:

```
/* measurement configuration */
meas_config_desc: {
a1_report_type: "rsrp",
a1_rsrp: -120,
a1_hysteresis: 0,
a1_time_to_trigger: 0,
a2_report_type: "rsrp",
a2_rsrp: -140,
a2_hysteresis: 0,
a2_time_to_trigger: 0,
a3_report_type: "rsrp",
a3_offset: -30,
a3_hysteresis: 0,
a3_time_to_trigger: 0,

scell_config: {
a2_report_type: "rsrp",
a2_rsrp: -140,
a2_hysteresis: 0,
a2_time_to_trigger: 0,
a4_report_type: "rsrp",
a4_rsrp: -120,
a4_hysteresis: 0,
a4_time_to_trigger: 0,
}
},
```

In this example the values of RSRP have been magnified in order to constantly receive measurement report from the device.

The measurement report introduction needs the insertation of `rrc_configuration: "measurement"` in the definition of the secondary cells list called `scell_list`, in this way the secondary cells (SCells) can be dynamically added and released based on measurement reports.

The neighbor cells are defined thanks to the array of objects called `scell_list`. Each neighbor object is defined by its identification and for each one of them it is possible to enable or not the Cross Carrier Scheduling feature. In case the cross-carrier scheduling is enabled, then the `scheduling_cell_id` needs to be specified. Each neighbor cell is also able to enable carrier aggregation in uplink by setting to true the `ul_allowed` boolean.

An example of code for the `cell_list` array is:

```
cell_list: [
   {
     rf_port: 0,
     cell_id: 0x01, /* low 8 bits of SIB1.cellIdentifier */
     tac: 0x0001, /* SIB1.trackingAreaCode */
     n_id_cell: 1,
     root_sequence_index: 204, /* PRACH root sequence index */

     /* carrier aggregation configuration (for rel 10 UEs) */
#if TDD == 1
   dl_earfcn: 40620, /* DL center frequency: 2593 MHz (band 41) */
#else
   dl_earfcn: 3350,  /* DL center frequency: 2680 MHz (Band 7) */
#endif

   /* list of secondary available cells */
   scell_list: [
    {
      cell_id: 0x02,
      ul_allowed: true,
      cross_carrier_scheduling: false,
//        cross_carrier_scheduling: true,
//        scheduling_cell_id: 0x01,
    },
    {
      cell_id: 0x03,
      ul_allowed: true,
      cross_carrier_scheduling: false,
//        cross_carrier_scheduling: true,
//        scheduling_cell_id: 0x01,
    },
   ],
  },
...
]
```

In the scenario C case, there will be two more cell objects of course.

5.3 - Measurement and performances

For the writing of this thesis SMS over IMS and VoLTE calls have been tested in basic scenarios such as:

- One cell SISO in FDD mode – scenario A.1
- One cell 2x2 MIMO in TDD mode – scenario A.2

- One cell 2x2 MIMO in FDD mode – scenario A.3

Handover, Carrier Aggregation, and Throughput have been tested in scenarios with higher complexity such as:

- Two cells MIMO in FDD mode – scenario B
- Three cells MIMO in FDD mode – scenario C

All the configuration files can be found in the Appendix of this thesis. The first three scenarios have in common the `N_CELL` equal to one given that they refer to cases with one cell only.

At the beginning of the eNB configuration file it is possible to find:

- `log_filename`, which sets the file name for the logs;
- `com_addr`, which is the address of the WebSocket server remote Application Programming Interface (API);
- `include "rf_driver/config.cfg"`, let the eNB know which SDR cards are being used;
- `rf_ports`, it is the array of the port objects;
- `mme_list`, it is the array of the mme objects to which the eNB is connected;
- `gtp_addr`, it is the IP address of the GTP. See chapter 2.1;
- `enb_id`, it is the identity of the eNB;
- `cell_list`, it is the array of the cell objects.

In case of simple scenarios with just one cell, the cell_list array contains one cell object only. A cell object is described by its EARFCN in downlink which defines the EARFCN in uplink automatically unless it is not otherwise specified. Each cell object is defined by its identification, Tracking Area Code (TAC), its PRACH root sequence index, its neighbor cells and the RF port the cell uses.

The configuration of the LTEENB continues with the definition of the parameters of the default cell inside the `cell_default` object, in this object it is possible to find:

- `plmn_list`, it is the array of PLMN objects;
- `uldl_config` and `sp_config`, these values are defined only in TDD case and they are used for the TDD frame configuration;

- `cyclic_prefix`, it is the guard period after each pulse carrying the modulated data symbol and it can be “normal” or “extended”;
- `phich_duration` and `phich_resource`, where PHICH stands for Physical Hybrid-ARQ Indicator channel and it carries ACKs/NACKs in DL to the UEs for the PUSCH received by the network;
- `si_value_tag`, `intra_freq_reselection`, `q_rx_lev_min`, and other parameters which define the System Information Block (SIB);
- `p_a` and `p_b`, both of them are useful parameters for the configuration of the Physical Downlink Shared Channel (PDSCH) which is shared by all UEs;
- `pdcch_format`, it is the number of CCEs for UE to 2^pdcch_format, if it is not set, the number of CCEs for the UE is computed from the CQI report;
- `prach_config_index` and `prach_freq_offset`, for the PRACH configuration index and the PRACH frequency offset;
- `simultaneousAckNackAndCQI_format3`, it enables simultaneous (N)ACK and CQI reporting with PUCCH format 3(r11);
- `pucch_dedicated` and `pusch_dedicated`, they are the objects for the PUCCH/PUSCH configuration;
- `initial_cqi`, it is the assumed CQI value when no value is received from the UE;
- `n_symb_cch`, `m_ri`, `transmission_mode`, `dl_256qam` and `ul_64qam`, see chapter 5.1;
- `ap_cqi_period` and `ap_cqi_rm`, period and reporting mode of the aperiodic CQI;
- `srs_dedicated`, `srb_config` and `mac_config`, they are the objects for the SRS/SRB/MAC configuration respectively;
- `dpc`, this optional boolean enables the dynamic power control, if it is set to true, there must be the definition for the SNR target for PUCCH and PUSCH;
- `cipher_algo_pref`, it sets the preferred algorithms for RRC and User Plane encryption.Then EEA0 (no encryption) is selected, if none match the UE capabilities;
- `integ_algo_pref`, Set the preferred algorithms for RRC integrity check. Then EIA0 (no integrity check) is selected, if none match the UE capabilities;

- `inactivity_timer`, after this time, in seconds, of inactivity it is sent the RRC connection release;
- `drb_config`, it includes the file for the Data Radio Bearer configuration.

As mentioned in the chapter 5.2, the device under test supports in uplink at maximum CA with 2 CCs. The following two figures show the obtained throughput in uplink in the scenario A.3 and the scenario B.

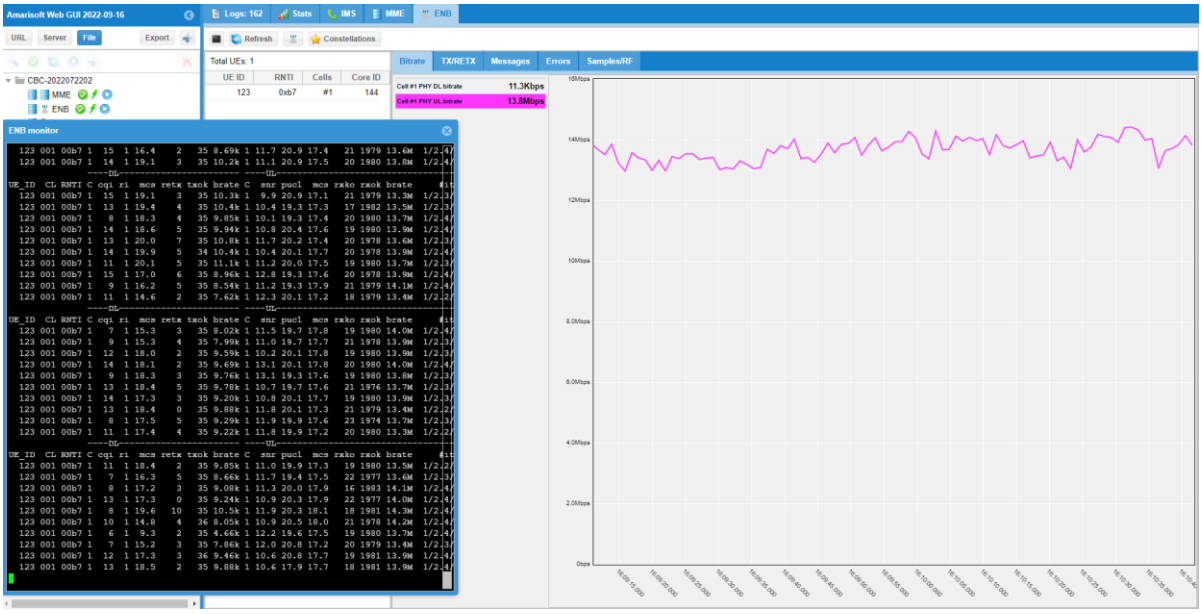


Figure 5.3.1: Throughput in Uplink in the scenario A.3

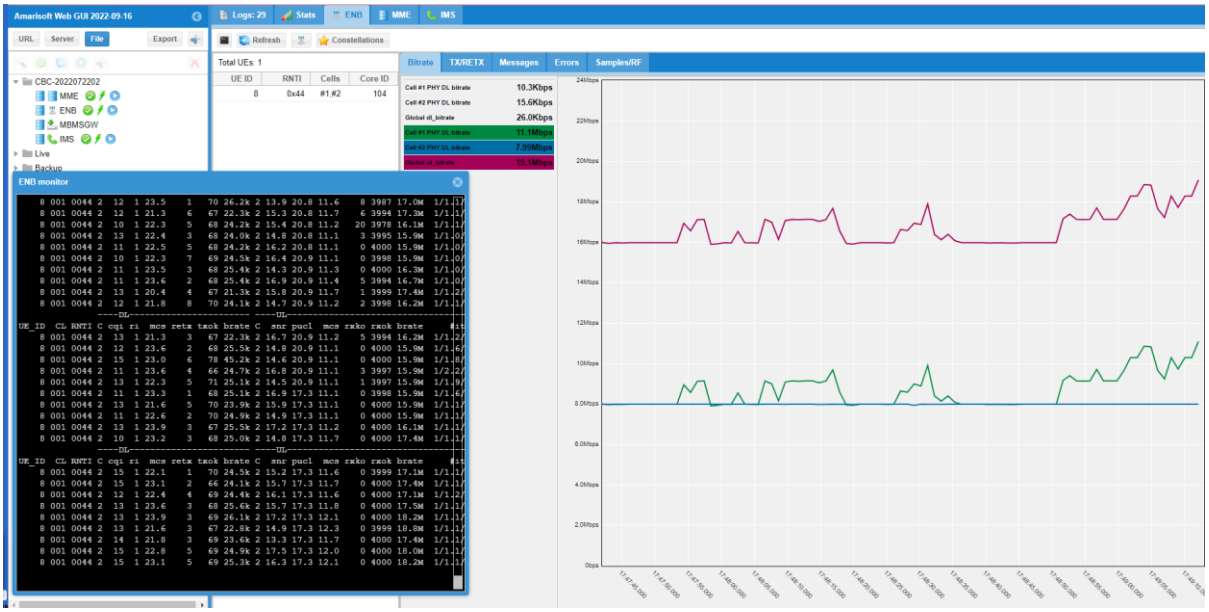


Figure 5.3.2: Throughput in Uplink in the scenario B

The scenario with one cell 2x2 MIMO in FDD mode gives an average throughput of 13 Mbps, this value increases in the scenario with two cells MIMO in FDD mode. In fact, the scenario B gives an average throughput of 17 Mbps, where each cell contributes with 8.5 Mbps roughly.

In downlink, the DUT supports Carrier Aggregation with up to three Carrier Components. The following figures exhibit the achieved throughput in downlink in the scenarios A.3, B and C.

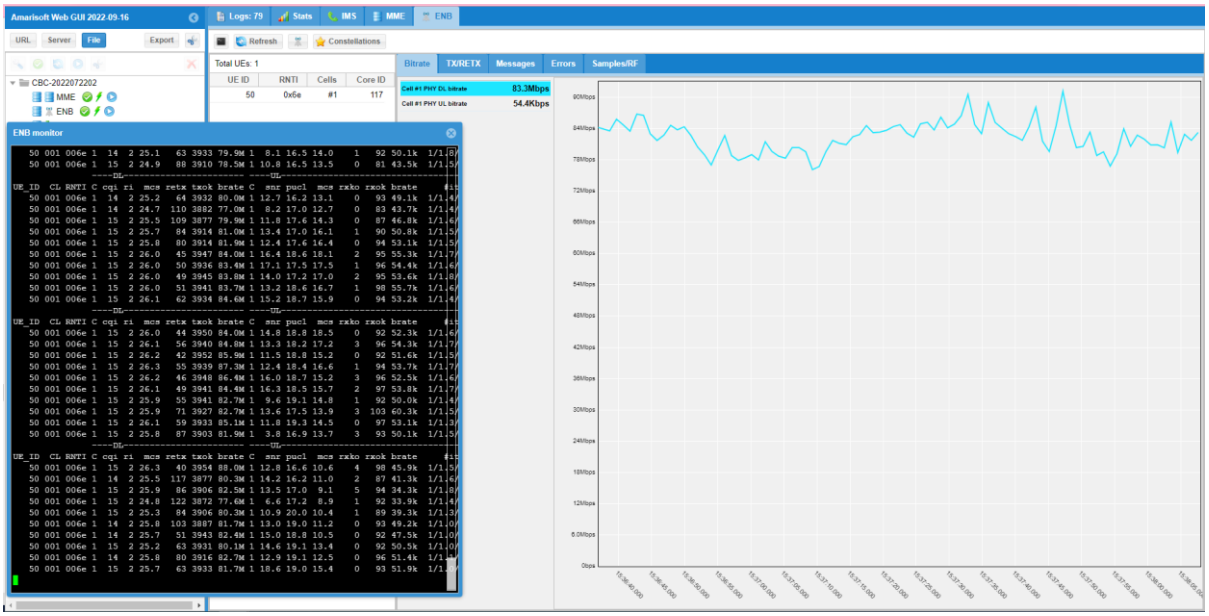


Figure 5.3.3: Throughput in Downlink in the scenario A.3

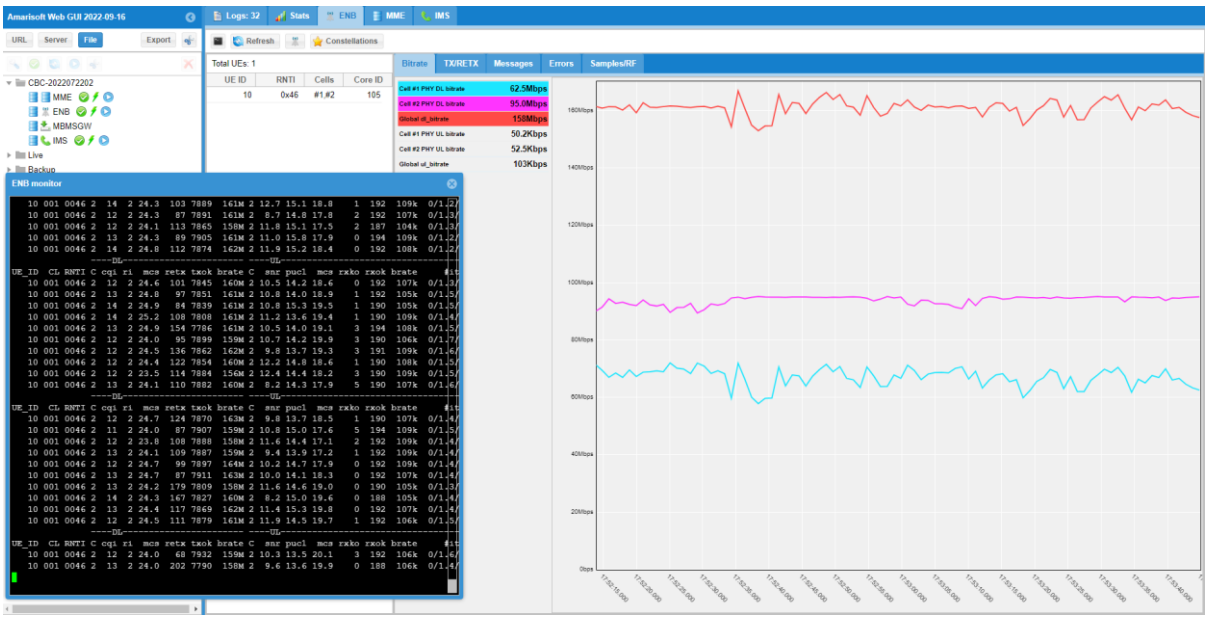


Figure 5.3.4: Throughput in Downlink in the scenario B

The scenario with one cell 2x2 MIMO in FDD mode (Figure 5.3.3) delivers an average throughput of 84 Mbps. The scenario with two cells MIMO in FDD mode (Figure 5.3.4) increases the average throughput to 160 Mbps. The cell one gives 70 Mbps roughly, while the cell two contributes with 90 Mbps approximately.

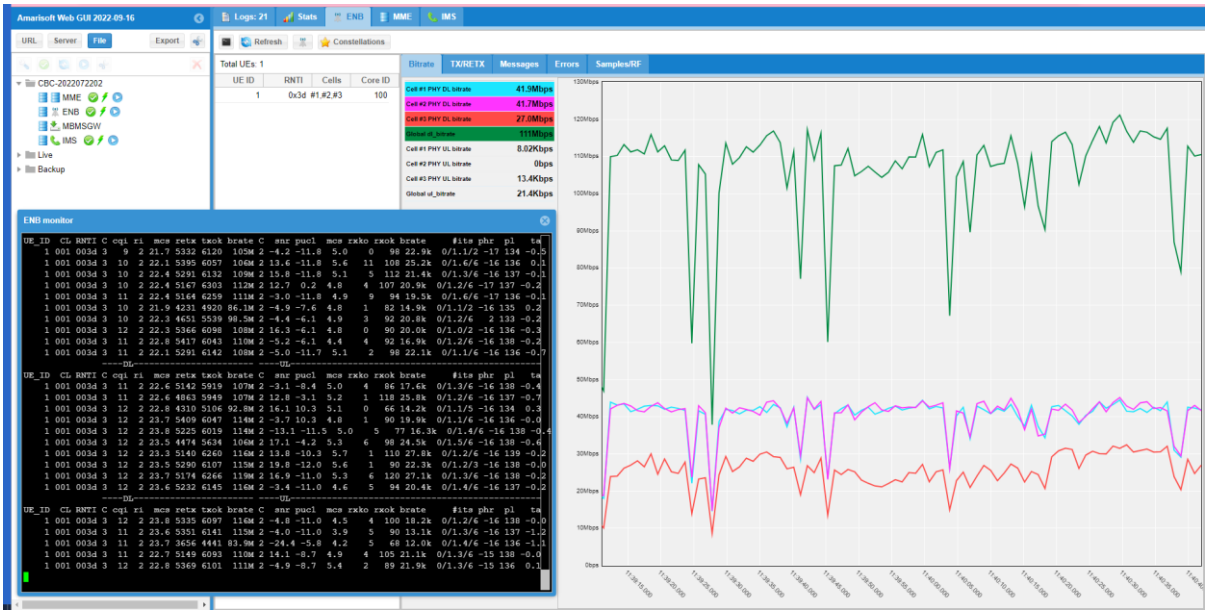


Figure 5.3.5: Throughput in Downlink in the scenario C

The scenario with three cells MIMO in FDD mode (Figure 5.3.5) provides an average throughput of 110 Mbps. In this scenario the first two cells give more or less the same bitrate while the third cell gives a lower bitrate. The reasons can be the choice of higher frequencies for cells one and two, and the higher distance between the UE and the MIMO antenna in the case of the cell three.

The peaks down happen because of the interference, given that all the measurements have been done in a non-static environment. Measuring the throughput in a real environment has not given the maximum values that the UE can achieve; nevertheless, comparing the obtained result with other MNOs' bitrates, 110 Mbps is still a good rate.

Blank page.

6 – Conclusion

The goal of this thesis was to deploy a private LTE mobile network in Open RAN, thanks to the virtualization approach of SDN. The challenges for this research were to enable Carrier Aggregation and to obtain satisfying Throughput values.

The analysis started with the study of the Evolved Packet System, the Open RAN architecture, and the Amarisoft architecture. Subsequently, Amari Callbox has been experimented by testing VoLTE calls, sending SMS messages, examining Handover between cells, enabling Carrier Aggregation, and measuring the Throughput.

The two challenges, mentioned at the beginning of this chapter, have been overpowered. Carrier Aggregation has, indeed, been enabled with up to three carrier components in downlink and two carrier components in uplink. For what concerns the Throughput, good results have been obtained.

Open RAN is ready for its commercial distribution, and it is the future of the mobile networks. The opening of the RAN will allow MNOs to benefit from intelligent networks, to avoid vendor lock-in and make any generation. For instance, it is remarkably accessible to emulate a private NR network starting from a private LTE network. Using Open RAN as the architecture for deployments in emerging markets will enable wider coverage at economical prices. In consequence, enterprises are trying to unblock the potential of Open RANs in order to make them efficient and secure. For example, MNOs are working on including Open RAN in the Network Equipment Security Assurance Scheme (NESAS), of the Global System for Mobile Communications Association (GSMA). For all these reasons, this thesis is certainly valued research.

The emulation of the private network in a real environment required a deep engagement with the network, allowing the exploitation of all the knowledge of the telecommunication master in order to face the obstacles met in the writing of this thesis. Nevertheless, the outcomes are gratifying.

Blank page.

Appendix

Scenario A.1 – One cell SISO FDD mode

++

./enb/config/enb-scenario-a1.cfg

++

```
/* lteenb configuration file version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */

#define N_CELL          1 // Values: 1 (one cell), 2 (two cells), 3 (three cells)
#define TDD             0 // Values: 0 (FDD), 1(TDD)
#define N_RB_DL                 25 // Values: 6 (1.4 MHz), 15 (3MHz), 25 (5MHz), 50
(10MHz), 75 (15MHz), 100 (20MHz)
#define N_ANTENNA_DL        1 // Values: 1 (SISO), 2 (MIMO 2x2)
#define N_ANTENNA_UL        1 // Values: 1, 2
#define CHANNEL_SIM         0 // Values: 0 (channel simulator disabled), 1 (channel
simulator enabled)
{
  /* Log filter: syntax: layer.field=value[,...]

     Possible layers are phy, mac, rlc, pdcp, rrc, nas, s1ap, x2ap, gtpu and
     all. The 'all' layer is used to address all the layers at the
     same time.

     field values:

     - 'level': the log level of each layer can be set to 'none',
     'error', 'info' or 'debug'. Use 'debug' to log all the messages.

     - 'max_size': set the maximum size of the hex dump. 0 means no
       hex dump. -1 means no limit.
  */
  //log_options: "all.level=debug,all.max_size=32",
  log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,s1ap.level=debug,s
1ap.max_size=1,x2ap.level=debug,x2ap.max_size=1,rrc.level=debug,rrc.max_size=
1",
  log_filename: "/tmp/enb0.log",

  /* Enable remote API and Web interface */
  com_addr: "0.0.0.0:9001",

  /* RF driver configuration */
  include "rf_driver/config.cfg",
```

```
#if CHANNEL_SIM == 1
  rf_ports: [
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    }
  ],
#endif

  mme_list: [
    {
      /* address of MME for S1AP connection. Must be modified if the MME
         runs on a different host. */
      mme_addr: "127.0.1.100",
    },
  ],
  /* GTP bind address (=address of the ethernet interface connected to
     the MME). Must be modified if the MME runs on a different host. */
  gtp_addr: "127.0.1.1",

  /* high 20 bits of SIB1.cellIdentifier */
  enb_id: 0x1A2D0,

  /* list of cells */
  cell_list: [
  {
    /* Broadcasted PLMN identities */
    plmn_list: [
      "00101",
    ],
#if TDD == 1
    //dl_earfcn: 38050, /* 2600 MHz (band 38) */
    dl_earfcn: 40620, /* 2593 MHz (band 41) */
    //dl_earfcn: 42590, /* 3500 MHz (band 42) */
#else
    //dl_earfcn: 300,   /* DL center frequency: 2132 MHz (Band 1) */
    //dl_earfcn: 900,   /* DL center frequency: 1960 MHz (Band 2) */
    //dl_earfcn: 1575,  /* DL center frequency: 1842.5 MHz (Band 3) */
    //dl_earfcn: 2150,  /* DL center frequency: 2130 MHz (Band 4) */
    //dl_earfcn: 2525,  /* DL center frequency: 881.5 MHz (Band 5) */
    dl_earfcn: 3350,  /* DL center frequency: 2680 MHz (Band 7) */
    //dl_earfcn: 6300,  /* 806 MHz (Band 20) */
    //dl_earfcn: 38050, /* 2600 MHz (band 38) */
    //dl_earfcn: 40620, /* 2593 MHz (band 41) */
    //dl_earfcn: 42590, /* 3500 MHz (band 42) */
#endif
```

```
    n_id_cell: 1,
    cell_id: 0x01,
    tac: 0x0001,
    root_sequence_index: 204, /* PRACH root sequence index */
  },
  ], /* cell_list */

  /* default cell parameters */
  cell_default: {
    n_antenna_dl: N_ANTENNA_DL, /* number of DL antennas */
    n_antenna_ul: N_ANTENNA_UL, /* number of UL antennas */

#if TDD == 1
    uldl_config: 2, /* TDD only */
    sp_config: 7, /* TDD only */
#endif

    n_rb_dl: N_RB_DL, /* Bandwidth: 25: 5 MHz, 50: 10 MHz, 75: 15 MHz, 100: 20 MHz
*/
    cyclic_prefix: "normal",

    phich_duration: "normal",
    phich_resource: "1", /* ratio of NG */

    /* SIB1 */
    si_value_tag: 0, /* increment modulo 32 if SI is modified */
    cell_barred: false, /* SIB1.cellBarred-r13 */
    intra_freq_reselection: true, /* SIB1.intraFreqReselection */
    q_rx_lev_min: -70, /* SIB1.q-RxLevMin */
    p_max: 10, /* maximum power allowed for the UE (dBm) */
    si_window_length: 40, /* ms */
    sib_sched_list: [
     {
       filename: "sib2_3.asn",
       si_periodicity: 16, /* frames */
     },
    ],

#if N_RB_DL == 6
    si_coderate: 0.30,   /* maximum code rate for SI/RA/P-RNTI messages */
#else
    si_coderate: 0.20,   /* maximum code rate for SI/RA/P-RNTI messages */
#endif
    si_pdcch_format: 2, /* 2 or 3. Log2 of the number of CCEs for PDCCH
                           for SI/RA/P-RNTI */

    n_symb_cch: 0, /* number of symbols for CCH (0 = auto) */

    /* PDSCH dedicated config (currently same for all UEs) */
    pdsch_dedicated: {
```

```
#if N_ANTENNA_DL == 4
      p_a: -6,
#elif N_ANTENNA_DL == 2
      p_a: -3,
#else
      p_a: 0,
#endif
      p_b: -1, /* -1 means automatic */
    },

    /* If defined, force for number of CCEs for UE specific PDCCH to
       2^pdcch_format. Otherwise it is computed from the reported
       CQI. Range: 0 to 3. */
#if N_RB_DL == 6
    pdcch_format: 1,
#else
    pdcch_format: 2,
#endif

    /* if defined, force the PDSCH MCS for all UEs. Otherwise it is
       computed from the reported CQI */
    /* pdsch_mcs: 12, */

#if N_RB_DL == 6
    prach_config_index: 15, /* subframe 9 every 20 ms */
#else
    prach_config_index: 4, /* subframe 4 every 10 ms */
#endif
    prach_freq_offset: -1, /* -1 means automatic */

    /* PUCCH dedicated config (currently same for all UEs) */
    pucch_dedicated: {
      n1_pucch_sr_count: 11, /* increase if more UEs are needed */
      cqi_pucch_n_rb: 1, /* increase if more UEs are needed */
#if TDD == 1
      //tdd_ack_nack_feedback_mode: "bundling", /* TDD only */
      tdd_ack_nack_feedback_mode: "multiplexing", /* TDD only */
#endif
    },

    /* PUSCH dedicated config (currently same for all UEs) */
    pusch_dedicated: {
      beta_offset_ack_index: 9,
      beta_offset_ri_index: 6,
      beta_offset_cqi_index: 6,
    },

    pusch_hopping_offset: -1, /* -1 means automatic */

    /* MCS for Msg3 (=CCCH RRC Connection Request) */
```

```
    pusch_msg3_mcs: 0,

    /* this CQI value is assumed when none is received from the UE */
#if N_RB_DL == 6
    initial_cqi: 5,
#else
    initial_cqi: 3,
#endif

    /* if defined, force the PUSCH MCS for all UEs. Otherwise it is
       computed from the last received SRS/PUSCH. */
    //  pusch_mcs: 18,

    dl_256qam: true,
    ul_64qam: true,

    /* Scheduling request period (ms). Must be >= 40 for HD-FDD */
    sr_period: 20,

    /* CQI report config */
    cqi_period: 40, /* period (ms). Must be >= 32 for HD-FDD */

#if N_ANTENNA_DL >= 2
    /* RI reporting is done with a period of m_ri * cqi_period.
       m_ri = 0 (default) disables RI reporting. */
    m_ri: 8,
    /* transmission mode */
    transmission_mode: 3,
#endif

    /* SRS dedicated config. All UEs share these
       parameters. srs_config_index and freq_domain_position are
       allocated for each UE) */
    srs_dedicated: {
#if N_RB_DL == 6
      srs_bandwidth_config: 7,
      srs_bandwidth: 1,
#elif N_RB_DL == 15
      srs_bandwidth_config: 6,
      srs_bandwidth: 1,
#elif N_RB_DL == 25
      srs_bandwidth_config: 3,
      srs_bandwidth: 1,
#elif N_RB_DL == 50
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#elif N_RB_DL == 75
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#else
```

```
    srs_bandwidth_config: 2,
    srs_bandwidth: 3,
#endif
    srs_subframe_config: 3, /* 0 - 15 */
    srs_period: 40, /* period (ms). Must be >= 40 for HD-FDD */
    srs_hopping_bandwidth: 0,
  },

  /* MAC configuration (same for all UEs) */
  mac_config: {
    ul_max_harq_tx: 5, /* max number of HARQ transmissions for uplink */
    dl_max_harq_tx: 5, /* max number of HARQ transmissions for downlink */
  },

  /* CPU load limitation */
  pusch_max_its: 6, /* max number of turbo decoder iterations */

  /* dynamic power control */
  dpc: true,
  dpc_pusch_snr_target: 25,
  dpc_pucch_snr_target: 20,

  /* RRC/UP ciphering algorithm preference. EEA0 is always the last. */
  cipher_algo_pref: [],
  /* RRC integrity algorithm preference. EIA0 is always the last. */
  integ_algo_pref: [2, 1],

  /* (in ms) send RRC connection release after this time of network
     inactivity */
  inactivity_timer: 10000,

  /* SRB configuration */
  srb_config: [
   {
    id: 1,
    maxRetxThreshold: 32,
    t_Reordering: 45,
    t_PollRetransmit: 60,
   },
   {
    id: 2 ,
    maxRetxThreshold: 32,
    t_Reordering: 45,
    t_PollRetransmit: 60,
   }
  ],

  /* DRB configuration */
  drb_config: "drb.cfg",
},
```

```
}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                                ./mme/config/mme.cfg

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/* ltemme configuration file for ims
 * version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */
{
  /* Log filter: syntax: layer.field=value[,...]
     Possible layers are nas, ip, s1ap, gtpu and all. The 'all' layer
     is used to address all the layers at the same time.
     field values:
     - 'level': the log level of each layer can be set to 'none',
     'error', 'info' or 'debug'. Use 'debug' to log all the messages.
     - 'max_size': set the maximum size of the hex dump. 0 means no
       hex dump. -1 means no limit.
  */
  //log_options: "all.level=debug,all.max_size=32",
log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,s1ap.level=debug,s
1ap.max_size=1,ngap.level=debug,ngap.max_size=1,rx.level=debug,rx.max_size=1,
cx.level=debug,cx.max_size=1",
  log_filename: "/tmp/mme.log",

  /* Enable remote API and Web interface */
  com_addr: "0.0.0.0:9000",

  /* bind address for GTP-U. Normally = address of the PC, here bound
    on local interface to be able to run ltemme on the same PC as
    lteenb. By default, the S1AP SCTP connection is bound on the same
    address. */
  gtp_addr: "127.0.1.100",

  plmn: "00101",
  mme_group_id: 32769,
  mme_code: 1,

  ims_vops_eps: true, /* IMS supported */
  ims_vops_5gs_3gpp: true, /* IMS supported */
  ims_vops_5gs_n3gpp: true, /* IMS supported */
  //emc_bs: true, /* emergency calls supported */
  //emc: 3, /* NR/E-UTRA connected to 5GCN emergency calls supported */
  //emc_n3gpp: true, /* non-3GPP emergency calls supported */
  emergency_number_list: [
     /* Category bits: (Table 10.5.135d/3GPP TS 24.008)
        Bit 1 Police
```

```
        Bit 2 Ambulance
        Bit 3 Fire Brigade
        Bit 4 Marine Guard
        Bit 5 Mountain Rescue
     */
   { category: 0x1f, digits: "911" },
   { category: 0x1f, digits: "112" },
],

rx: {
  qci: {audio: 1, video: 2},
},

/* network name and network short name sent in the EMM information
   message to the UE */
network_name: "Telebit Network",
network_short_name: "TelebitNET",
//network_name: "Amarisoft Network",
//network_short_name: "Amarisoft",

/* Control Plane Cellular IoT EPS optimization support */
cp_ciot_opt: true,

/* DCNR support */
dcnr_support: true,

eps_5gs_interworking: "with_n26",

/* 15 bearers support */
fifteen_bearers: false,
ims_list: [{ims_addr: "127.0.0.1", bind_addr: "127.0.0.2"}],

/* AMF slices configuration */
/*nssai: [
 {
  sst: 1,
 },
 {
  sst: 2,
 },
 {
  sst: 3,
  sd: 50,
 }
],*/
/* ePDG configuration */
//epdg: {
//  bind_addr: "127.0.1.100:500",
//  esp_duration: 900,
//  certificate: "epdg.pem",
```

```
//  /* required for some buggy Mediatek phones */
//  //omit_auth_in_first_auth_rsp: true
//},
/* Public Data Networks. The first one is the default. */
pdn_list: [
 {
   pdn_type: "ipv4",
   access_point_name: "default",
   first_ip_addr: "192.168.2.2",
   last_ip_addr: "192.168.2.254",
   ip_addr_shift: 2, /* difference between allocated IP addresses is 4 */
   dns_addr: "8.8.8.8", /* Google DNS address */

   erabs: [
    {
      qci: 9,
      priority_level: 15,
      pre_emption_capability: "shall_not_trigger_pre_emption",
      pre_emption_vulnerability: "not_pre_emptable",
    },
   ],
 },
 {
   pdn_type: "ipv4",
   access_point_name: "internet",
   first_ip_addr: "192.168.3.2",
   last_ip_addr: "192.168.3.254",
   ip_addr_shift: 2, /* difference between allocated IP addresses is 4 */
   dns_addr: "8.8.8.8", /* Google DNS address */

   /* IPv6 sample config
   pdn_type: "ipv4v6",
   first_ipv6_prefix: "2001:468:2000:1::",
   last_ipv6_prefix: "2001:468:2000:ffff::",
   dns_addr: ["8.8.8.8", "2001:4860:4860::8888"], // Google IPv6 DNS address
   */

   erabs: [
    {
      qci: 9,
      priority_level: 15,
      pre_emption_capability: "shall_not_trigger_pre_emption",
      pre_emption_vulnerability: "not_pre_emptable",
    },
   ],

   /*slices: [
   {
    snssai: {
     sst: 1,
```

```
    },
    qos_flows: [
     {
      "5qi": 6,
       priority_level: 9,
       pre_emption_capability: "shall_not_trigger_pre_emption",
       pre_emption_vulnerability: "not_pre_emptable",
     },
    ],
   },
   {
    snssai: {
     sst: 3,
     sd: 50,
    },
    qos_flows: [
     {
      "5qi": 7,
       priority_level: 8,
       pre_emption_capability: "shall_not_trigger_pre_emption",
       pre_emption_vulnerability: "not_pre_emptable",
     },
    ],
   }
  ],*/
 },
 {
  access_point_name: "ims",
  pdn_type: "ipv4v6",
  first_ip_addr: "192.168.4.2",
  last_ip_addr: "192.168.4.254",
  ip_addr_shift: 2, /* difference between allocated IP addresses is 4 */
  first_ipv6_prefix: "2001:468:3000:1::",
  last_ipv6_prefix: "2001:468:3000:ffff::",
  p_cscf_addr: ["192.168.4.1", "2001:468:3000:1::"],
  dns_addr: ["8.8.8.8", "2001:4860:4860::8888"], // Google IPv6 DNS address

  erabs: [
   {
    qci: 5,
    priority_level: 15,
    pre_emption_capability: "shall_not_trigger_pre_emption",
    pre_emption_vulnerability: "not_pre_emptable",
   },
  ], /* erabs */
 },
 {
  access_point_name: "sos",
  emergency: true,
  pdn_type: "ipv4v6",
```

```
    first_ip_addr: "192.168.5.2",
    last_ip_addr: "192.168.5.254",
    ip_addr_shift: 2, /* difference between allocated IP addresses is 4 */
    first_ipv6_prefix: "2001:468:4000:1::",
    last_ipv6_prefix: "2001:468:4000:ffff::",
    p_cscf_addr: ["192.168.5.1", "2001:468:4000:1::"],
    dns_addr: ["8.8.8.8", "2001:4860:4860::8888"], // Google IPv6 DNS address

    erabs: [
     {
       qci: 5,
       priority_level: 15,
       pre_emption_capability: "shall_not_trigger_pre_emption",
       pre_emption_vulnerability: "not_pre_emptable",
     },
    ], /* erabs */
  }
 ],

 /* Setup script for the network interface.
    If no script is given, no network interface is created.
    Script is called for each PDN with following parameters:
       1) Interface name
       2) PDN index
       3) Access Point Name
       4) IP version: 'ipv4' or 'ipv6'
       5) IP address: first IP address for ipv4 and link local address for IPv6
       6) First IP address
       7) Last IP address
 */
 tun_setup_script: "mme-ifup",

 /* NAS ciphering algorithm preference. EEA0 is always the last. */
 nas_cipher_algo_pref: [ ],
 /* NAS integrity algorithm preference. EIA0 is always the last. */
 nas_integ_algo_pref: [ 2, 1 ],

 /* user data base */
 include "ue_db-ims.cfg",
}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

./mme/config/ims.cfg

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/* lteims configuration file
 * version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */
```

```
{
   //log_options: "all.level=debug,all.max_size=32",
   log_options:
"all.level=error,sip.level=debug,sip.max_size=1,ims.level=debug,rx.level=debug,rx.max_size=1,cx.level=debug,cx.max_size=1",
   log_filename: "/tmp/ims.log",

   /* SIP bind address */
   sip_addr: [
      {addr: "192.168.4.1", bind_addr: "192.168.4.1", port_min: 10000, port_max: 20000},
      "2001:468:3000:1::",
      {addr: "192.168.5.1", bind_addr: "192.168.5.1", port_min: 10000, port_max: 20000},
      "2001:468:4000:1::"
   ],

   mms_server_bind_addr: "192.168.3.1:1111",

   /* MME connection for SMS over SG */
   sctp_addr: "127.0.0.1",

   /* Cx connection */
   cx_server_addr: "127.0.1.100",
   cx_bind_addr: "127.0.0.1",

   /* Rx connection */
   rx_server_addr: "127.0.1.100",

   /* Remote API */
   com_addr: "0.0.0.0:9003",

   /* Global domain name (May be overriden for each user) */
   domain: "amarisoft.com",

   /* user data base */
   include "ue_db-ims.cfg",

   /* Echo phone number */
   echo: [
      "tel:666",
      "tel:+666",
      {impu: "tel:404", code: 404},     /* 404 test */
      {impu: "urn:service:sos", anonymous: true, authentication: false}, /* Emergency call */
      {impu: "urn:service:sos.police", anonymous: true, authentication: false}, /* Emergency call */
   ],

   /* Delay */
```

```
    //sms_expires: 86400,
    //binding_expires: 3600,

    /* on: 3GPP mode allowed
     * silent: 3GPP mode forced
     * off: IETF mode
     */
    precondition: "on",
    "100rel": true,

    /* IPsec */
    ipsec_aalg_list: ["hmac-md5-96", "hmac-sha-1-96"],
    ipsec_ealg_list: ["null", "aes-cbc", "des-cbc", "des-ede3-cbc"],

    mt_call_sdp_file: "mt_call_qos.sdp",

    ue_db_filename: "lte_ue_ims.db",
}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

./mme/config/ue_db-ims.cfg

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ue_db: [{
    sim_algo: "xor", /* USIM authentication algorithm: xor, milenage or tuak */
    imsi: "001010123456789", /* Anritsu Test USIM */
//    imsi: "001012345678901", /* Agilent or R&S Test USIM */
    amf: 0x9001, /* Authentication Management Field */
    sqn: "000000000000", /* Sequence Number */
    K: "00112233445566778899aabbccddeeff", /* Anritsu Test USIM */
//    K: "4147494C454E5420544543484E4F0000", /* Agilent Test USIM */
//    K: "000102030405060708090A0B0C0D0E0F", /* R&S Test USIM */

    impi: "001010123456789@ims.mnc001.mcc001.3gppnetwork.org",
    impu: [
        "001010123456789",
          {impu: "tel:0601", imei:"866929050364853"},
          {impu: "tel:0602", imei:"866929050363731"},
    ],
    domain: "ims.mnc001.mcc001.3gppnetwork.org",
    multi_sim: true, /* Experimental */

    /* For standard SIP client */
    /*pwd:  "amarisoft",
    authent_type: "MD5",*/
}, {
    sim_algo: "milenage",
    imsi: "001010000000001",
    opc: "000102030405060708090A0B0C0D0E0F",
```

```
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: ["001010000000001", "tel:0600000001"],
    impi: "001010000000001@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000002",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: ["001010000000002", "tel:0600000002"],
    impi: "001010000000002@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000003",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: ["001010000000003", "tel:0600000003"],
    impi: "001010000000003@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000004",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000004",
    impi: "001010000000004@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000005",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000005",
    impi: "001010000000005@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000006",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000006",
    impi: "001010000000006@ims.mnc001.mcc001.3gppnetwork.org",
```

```
}, {
    sim_algo: "milenage",
    imsi: "001010000000007",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000007",
    impi: "001010000000007@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000008",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000008",
    impi: "001010000000008@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000009",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000009",
    impi: "001010000000009@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000010",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000010",
    impi: "001010000000010@ims.mnc001.mcc001.3gppnetwork.org",
}]
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

./enb/config/rf_driver/config.cfg

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/* Parameters for SDR device version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */
#if !defined(N_CELL)
  #define N_CELL 1
#endif
```

```
#if !defined(TDD)
  #define TDD 0
#endif

#if !defined(N_ANTENNA_DL)
  #define N_ANTENNA_DL 1
#endif

#if N_ANTENNA_DL <= 2
  #define N_CHAN N_CELL
#else
  #define N_CHAN 2*N_CELL
#endif

rf_driver: {
    name: "sdr",
    /* list of devices. 'dev0' is always the master. */
#if N_CHAN == 1
    args: "dev0=/dev/sdr0",
#elif N_CHAN == 2
    args: "dev0=/dev/sdr0,dev1=/dev/sdr1",
#elif N_CHAN == 3
    args: "dev0=/dev/sdr0,dev1=/dev/sdr1,dev2=/dev/sdr2",
#elif N_CHAN == 4
    args: "dev0=/dev/sdr0,dev1=/dev/sdr1,dev2=/dev/sdr2,dev3=/dev/sdr3",
#elif N_CHAN == 6
args:"dev0=/dev/sdr0,dev1=/dev/sdr1,dev2=/dev/sdr2,dev3=/dev/sdr3,dev4=/dev/sdr
4,dev5=/dev/sdr5",
#else
    args: "",
#endif

    /* synchronisation source: none, internal, gps, external (default = none) */
    // sync: "gps",
#if TDD == 1
    rx_antenna:"auto", // force to use the RX connector in TDD as RX antenna
#endif

    /* PCIe jitter. decrease it to reduce latency */
    //fifo_tx_time: 10,
    //rx_latency: 30,
},

#if 0
  tx_pad_duration:300,
#endif

tx_gain: 90.0, /* TX gain (in dB) */
rx_gain: 60.0, /* RX gain (in dB) */
```

++

./ots/config/ots.cfg

++

```
# Copyright (C) 2012-2022 Amarisoft
#
# LTE service default configuration 2022-09-16
#
# Please do not edit this file as it will be replaced on next install
#
# You may create your own config file and include this one with source shell
# command.
# Then, update ots.cfg symbolink link to point to this file.
# Ex:
# echo "source ots.default.cfg" > my-ots.cfg
# rm ots.cfg && ln -s my-ots.cfg
# Then put your custom config in my-ots.cfg
# And restart lte service: "service lte restart"

# General
ERROR_DELAY="5"    # Component restart time in case of error (in seconds)
#AMARISOFT_PATH=""  # Change license file location
#HOSTNAME="My name" # Use this to override system hostname (| are forbidden)

# Logs
LOG_FILE="ots.log"
LOG_FILE_SIZE="1M" # Service log file size threshold for rotation
LOG_SIZE="250M" # Components log file size threshold for rotation
LOG_PATH="/var/log/lte/" # Log rotation target path
LOG_PERSISTENT_SIZE="5G" # Maximum size of logs to store in LOG_PATH (if no unit KBytes assumed)
LOG_PERSISTENT_COUNT="2000" # Maximum number of log file to keep in LOG_PATH
LOG_GZIP="0" # Set to positive value to compress logs in LOG_PATH
LOG_POLL_DELAY="10"
LOG_BACKUP_ON_STOP="y" # Set it to n to avoid log backup (to LOG_PATH) on component stop

# Component configuration
# <COMP> is component ID and must be unique
#
# <COMP>_TYPE: component type (LICENSE MME ENB IMS MBMSGW UE...)
# <COMP>_TITLE: display name
# <COMP>_PATH: Component path. Set it to empty string to disable component
# <COMP>_INIT: arguments for lte_init.sh script
# <COMP>_WIN: Component screen window # (must be unique)
# <COMP>_CONFIG_FILE: config file used by component
# <COMP>_OUTPUT_FILE: define it to dump stdout/stderr to a file
```

```
# <COMP>_AUTOSTART: defines if component must be started by service (default is y)
# <COMP>_SCRIPT: defines shell script that will be executed on each component state change
#               Arguments are <COMP> <STATE> [<ERROR>] where state can be:
#               - starting: before executing component binary
#               - started: after component binary has started
#                 - error: when any error occurred (In this case the third argument will be filed with error message)
#               - stop: after successful component stop
#                This can be used to perform any action before component start by using the state starting
# <COMP>_START_DELAY: time to wait in seconds before starting component

# List of components to start
COMPONENTS=""

# Start of section generated by installer
# Mon Oct 10 12:37:35 PM UTC 2022

# LTE automatic service config
OTS_PATH="/root/ots"
#OTS_AUTOSTART="y"
WWW_PATH="/var/www/html/lte/"

# EPC config
COMPONENTS+=" MME"
MME_TYPE="MME"
MME_WIN="0"
MME_PATH="/root/mme"
#MME_AUTOSTART="y"
MME_INIT=" -6"
MME_CONFIG_FILE="config/mme.cfg"

# IMS config
COMPONENTS+=" IMS"
IMS_TYPE="IMS"
IMS_WIN="3"
IMS_PATH="/root/mme"
#IMS_AUTOSTART="y"
IMS_DEP="MME"
IMS_CONFIG_FILE="config/ims.cfg"

# eNB config
COMPONENTS+=" ENB"
ENB_TYPE="ENB"
ENB_WIN="1"
ENB_PATH="/root/enb"
#ENB_AUTOSTART="y"
ENB_INIT=""
```

```
ENB_RRH_CHECK="config/rf_driver/rrh_check.sh --ots"
ENB_CONFIG_FILE="config/enb.cfg"

# MBMS gateway config
COMPONENTS+=" MBMSGW"
MBMSGW_TYPE="MBMSGW"
MBMSGW_WIN="4"
MBMSGW_PATH="/root/mbms"
#MBMSGW_AUTOSTART="y"
MBMSGW_INIT=""
MBMSGW_CONFIG_FILE="config/mbmsgw.cfg"

# System configuration
HT_STATE="off"

# End of section generated by installer
```

Scenario A.2 – One cell 2x2 MIMO TDD mode

++

./enb/config/enb-scenario-a2.cfg

++

```
/* lteenb configuration file version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */

#define N_CELL          1
#define TDD             1 // Values: 0 (FDD), 1(TDD)
#define N_RB_DL         25 // Values: 6 (1.4 MHz), 15 (3MHz), 25 (5MHz), 50 (10MHz), 75 (15MHz), 100 (20MHz)
#define N_ANTENNA_DL    2 // Values: 1 (SISO), 2 (MIMO 2x2)
#define N_ANTENNA_UL    2 // Values: 1, 2
#define CHANNEL_SIM     0 // Values: 0 (channel simulator disabled), 1 (channel simulator enabled)
#define NG_ENB          0 // 1 for ng-eNB

{
  /* Log filter: syntax: layer.field=value[,...]

     Possible layers are phy, mac, rlc, pdcp, rrc, nas, s1ap, x2ap, gtpu and
     all. The 'all' layer is used to address all the layers at the
     same time.

     field values:

     - 'level': the log level of each layer can be set to 'none',
     'error', 'info' or 'debug'. Use 'debug' to log all the messages.

     - 'max_size': set the maximum size of the hex dump. 0 means no
       hex dump. -1 means no limit.
  */
  //log_options: "all.level=debug,all.max_size=32",
  log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,s1ap.level=debug,s1ap.max_size=1,x2ap.level=debug,x2ap.max_size=1,rrc.level=debug,rrc.max_size=1",
  log_filename: "/tmp/enb0.log",

  /* Enable remote API and Web interface */
  com_addr: "0.0.0.0:9001",

  /* RF driver configuration */
  include "rf_driver/config.cfg",
```

```
#if CHANNEL_SIM == 1
  rf_ports: [
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    }
  ],
#endif

  mme_list: [
    {
      /* address of MME for S1AP connection. Must be modified if the MME
         runs on a different host. */
      mme_addr: "127.0.1.100",
    },
  ],
#if NG_ENB == 1
  amf_list: [
    {
      /* address of AMF for NGAP connection. Must be modified if the AMF
         runs on a different host. */
      amf_addr: "127.0.1.100",
    },
  ],
#endif
  /* GTP bind address (=address of the ethernet interface connected to
     the MME). Must be modified if the MME runs on a different host. */
  gtp_addr: "127.0.1.1",

  /* high 20 bits of SIB1.cellIdentifier */
  enb_id: 0x1A2D0,

  /* list of cells */
  cell_list: [
  {
    /* Broadcasted PLMN identities */
    plmn_list: [
      "00101",
    ],
#if NG_ENB == 1
    plmn_list_5gc: [ {
      tac: 10,
      plmn_ids: [{ plmn: "00101", reserved: false }],
    }],
#endif

#if TDD == 1
    //dl_earfcn: 38050, /* 2600 MHz (band 38) */
```

```
    dl_earfcn: 40620, /* 2593 MHz (band 41) */
    //dl_earfcn: 42590, /* 3500 MHz (band 42) */
#else
    //dl_earfcn: 300,   /* DL center frequency: 2132 MHz (Band 1) */
    //dl_earfcn: 900,   /* DL center frequency: 1960 MHz (Band 2) */
    //dl_earfcn: 1575,  /* DL center frequency: 1842.5 MHz (Band 3) */
    //dl_earfcn: 2150,  /* DL center frequency: 2130 MHz (Band 4) */
    //dl_earfcn: 2525,  /* DL center frequency: 881.5 MHz (Band 5) */
    dl_earfcn: 3350,  /* DL center frequency: 2680 MHz (Band 7) */
    //dl_earfcn: 6300,  /* 806 MHz (Band 20) */
    //dl_earfcn: 38050, /* 2600 MHz (band 38) */
    //dl_earfcn: 40620, /* 2593 MHz (band 41) */
    //dl_earfcn: 42590, /* 3500 MHz (band 42) */
#endif

    n_id_cell: 1,
    cell_id: 0x01,
    tac: 0x0001,
    root_sequence_index: 204, /* PRACH root sequence index */
  },
  ], /* cell_list */

  /* default cell parameters */
  cell_default: {
    n_antenna_dl: N_ANTENNA_DL, /* number of DL antennas */
    n_antenna_ul: N_ANTENNA_UL, /* number of UL antennas */

#if TDD == 1
    uldl_config: 2, /* TDD only */
    sp_config: 7, /* TDD only */
#endif

    n_rb_dl: N_RB_DL, /* Bandwidth: 25: 5 MHz, 50: 10 MHz, 75: 15 MHz, 100: 20 MHz
*/
    cyclic_prefix: "normal",

    phich_duration: "normal",
    phich_resource: "1", /* ratio of NG */

    /* SIB1 */
    si_value_tag: 0, /* increment modulo 32 if SI is modified */
    cell_barred: false, /* SIB1.cellBarred-r13 */
    intra_freq_reselection: true, /* SIB1.intraFreqReselection */
    q_rx_lev_min: -70, /* SIB1.q-RxLevMin */
    p_max: 10, /* maximum power allowed for the UE (dBm) */
    si_window_length: 40, /* ms */
    sib_sched_list: [
     {
       filename: "sib2_3.asn",
       si_periodicity: 16, /* frames */
```

```
      },
    ],

#if N_RB_DL == 6
    si_coderate: 0.30,   /* maximum code rate for SI/RA/P-RNTI messages */
#else
    si_coderate: 0.20,   /* maximum code rate for SI/RA/P-RNTI messages */
#endif
    si_pdcch_format: 2, /* 2 or 3. Log2 of the number of CCEs for PDCCH
                           for SI/RA/P-RNTI */

    n_symb_cch: 0, /* number of symbols for CCH (0 = auto) */

    /* PDSCH dedicated config (currently same for all UEs) */
    pdsch_dedicated: {
#if N_ANTENNA_DL == 4
      p_a: -6,
#elif N_ANTENNA_DL == 2
      p_a: -3,
#else
      p_a: 0,
#endif
      p_b: -1, /* -1 means automatic */
    },

    /* If defined, force for number of CCEs for UE specific PDCCH to
       2^pdcch_format. Otherwise it is computed from the reported
       CQI. Range: 0 to 3. */
#if N_RB_DL == 6
    pdcch_format: 1,
#else
    pdcch_format: 2,
#endif

    /* if defined, force the PDSCH MCS for all UEs. Otherwise it is
       computed from the reported CQI */
    /* pdsch_mcs: 12, */

#if N_RB_DL == 6
    prach_config_index: 15, /* subframe 9 every 20 ms */
#else
    prach_config_index: 4, /* subframe 4 every 10 ms */
#endif
    prach_freq_offset: -1, /* -1 means automatic */

    /* PUCCH dedicated config (currently same for all UEs) */
    pucch_dedicated: {
      n1_pucch_sr_count: 11, /* increase if more UEs are needed */
      cqi_pucch_n_rb: 1, /* increase if more UEs are needed */
#if TDD == 1
```

```
      //tdd_ack_nack_feedback_mode: "bundling", /* TDD only */
      tdd_ack_nack_feedback_mode: "multiplexing", /* TDD only */
#endif
    },

    /* PUSCH dedicated config (currently same for all UEs) */
    pusch_dedicated: {
      beta_offset_ack_index: 9,
      beta_offset_ri_index: 6,
      beta_offset_cqi_index: 6,
    },

    pusch_hopping_offset: -1, /* -1 means automatic */

    /* MCS for Msg3 (=CCCH RRC Connection Request) */
    pusch_msg3_mcs: 0,

    /* this CQI value is assumed when none is received from the UE */
#if N_RB_DL == 6
    initial_cqi: 5,
#else
    initial_cqi: 3,
#endif

    /* if defined, force the PUSCH MCS for all UEs. Otherwise it is
       computed from the last received SRS/PUSCH. */
    //  pusch_mcs: 18,

    dl_256qam: true,
    ul_64qam: true,

    /* Scheduling request period (ms). Must be >= 40 for HD-FDD */
    sr_period: 20,

    /* CQI report config */
    cqi_period: 40, /* period (ms). Must be >= 32 for HD-FDD */

#if N_ANTENNA_DL >= 2
    /* RI reporting is done with a period of m_ri * cqi_period.
       m_ri = 0 (default) disables RI reporting. */
    m_ri: 8,
    /* transmission mode */
    transmission_mode: 3,
#endif

    /* SRS dedicated config. All UEs share these
       parameters. srs_config_index and freq_domain_position are
       allocated for each UE) */
    srs_dedicated: {
#if N_RB_DL == 6
```

```
      srs_bandwidth_config: 7,
      srs_bandwidth: 1,
#elif N_RB_DL == 15
      srs_bandwidth_config: 6,
      srs_bandwidth: 1,
#elif N_RB_DL == 25
      srs_bandwidth_config: 3,
      srs_bandwidth: 1,
#elif N_RB_DL == 50
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#elif N_RB_DL == 75
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#else
      srs_bandwidth_config: 2,
      srs_bandwidth: 3,
#endif
      srs_subframe_config: 3, /* 0 - 15 */
      srs_period: 40, /* period (ms). Must be >= 40 for HD-FDD */
      srs_hopping_bandwidth: 0,
    },

    /* MAC configuration (same for all UEs) */
    mac_config: {
      ul_max_harq_tx: 5, /* max number of HARQ transmissions for uplink */
      dl_max_harq_tx: 5, /* max number of HARQ transmissions for downlink */
    },

    /* CPU load limitation */
    pusch_max_its: 6, /* max number of turbo decoder iterations */

    /* dynamic power control */
    dpc: true,
    dpc_pusch_snr_target: 25,
    dpc_pucch_snr_target: 20,

    /* RRC/UP ciphering algorithm preference. EEA0 is always the last. */
    cipher_algo_pref: [],
    /* RRC integrity algorithm preference. EIA0 is always the last. */
    integ_algo_pref: [2, 1],

    /* (in ms) send RRC connection release after this time of network
       inactivity */
    inactivity_timer: 10000,

    /* SRB configuration */
    srb_config: [
     {
       id: 1,
```

```
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      },
      {
        id: 2 ,
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      }
    ],

    /* DRB configuration */
    drb_config: "drb.cfg",
  },
}
```

Scenario A.3 – One cell 2x2 MIMO FDD mode

++

./enb/config/enb-scenario-a3.cfg

++

```
/* lteenb configuration file version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */

#define N_CELL          1
#define TDD             0 // Values: 0 (FDD), 1(TDD)
#define N_RB_DL         25 // Values: 6 (1.4 MHz), 15 (3MHz), 25 (5MHz), 50 (10MHz), 75 (15MHz), 100 (20MHz)
#define N_ANTENNA_DL    2 // Values: 1 (SISO), 2 (MIMO 2x2)
#define N_ANTENNA_UL    2 // Values: 1, 2
#define CHANNEL_SIM     0 // Values: 0 (channel simulator disabled), 1 (channel simulator enabled)
#define NG_ENB          0 // 1 for ng-eNB

{
  /* Log filter: syntax: layer.field=value[,...]

     Possible layers are phy, mac, rlc, pdcp, rrc, nas, s1ap, x2ap, gtpu and
     all. The 'all' layer is used to address all the layers at the
     same time.

     field values:

     - 'level': the log level of each layer can be set to 'none',
     'error', 'info' or 'debug'. Use 'debug' to log all the messages.

     - 'max_size': set the maximum size of the hex dump. 0 means no
       hex dump. -1 means no limit.
  */
  //log_options: "all.level=debug,all.max_size=32",
  log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,s1ap.level=debug,s1ap.max_size=1,x2ap.level=debug,x2ap.max_size=1,rrc.level=debug,rrc.max_size=1",
  log_filename: "/tmp/enb0.log",

  /* Enable remote API and Web interface */
  com_addr: "0.0.0.0:9001",

  /* RF driver configuration */
  include "rf_driver/config.cfg",
```

```
#if CHANNEL_SIM == 1
  rf_ports: [
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    }
  ],
#endif

  mme_list: [
    {
      /* address of MME for S1AP connection. Must be modified if the MME
         runs on a different host. */
      mme_addr: "127.0.1.100",
    },
  ],
#if NG_ENB == 1
  amf_list: [
    {
      /* address of AMF for NGAP connection. Must be modified if the AMF
         runs on a different host. */
      amf_addr: "127.0.1.100",
    },
  ],
#endif
  /* GTP bind address (=address of the ethernet interface connected to
     the MME). Must be modified if the MME runs on a different host. */
  gtp_addr: "127.0.1.1",

  /* high 20 bits of SIB1.cellIdentifier */
  enb_id: 0x1A2D0,

  /* list of cells */
  cell_list: [
  {
    /* Broadcasted PLMN identities */
    plmn_list: [
      "00101",
    ],
#if NG_ENB == 1
    plmn_list_5gc: [ {
      tac: 10,
      plmn_ids: [{ plmn: "00101", reserved: false }],
    }],
#endif

#if TDD == 1
    //dl_earfcn: 38050, /* 2600 MHz (band 38) */
```

```
    dl_earfcn: 40620, /* 2593 MHz (band 41) */
    //dl_earfcn: 42590, /* 3500 MHz (band 42) */
#else
    //dl_earfcn: 300,   /* DL center frequency: 2132 MHz (Band 1) */
    //dl_earfcn: 900,   /* DL center frequency: 1960 MHz (Band 2) */
    //dl_earfcn: 1575,  /* DL center frequency: 1842.5 MHz (Band 3) */
    //dl_earfcn: 2150,  /* DL center frequency: 2130 MHz (Band 4) */
    //dl_earfcn: 2525,  /* DL center frequency: 881.5 MHz (Band 5) */
    dl_earfcn: 3350,  /* DL center frequency: 2680 MHz (Band 7) */
    //dl_earfcn: 6300,  /* 806 MHz (Band 20) */
    //dl_earfcn: 38050, /* 2600 MHz (band 38) */
    //dl_earfcn: 40620, /* 2593 MHz (band 41) */
    //dl_earfcn: 42590, /* 3500 MHz (band 42) */
#endif

    n_id_cell: 1,
    cell_id: 0x01,
    tac: 0x0001,
    root_sequence_index: 204, /* PRACH root sequence index */
  },
  ], /* cell_list */

  /* default cell parameters */
  cell_default: {
    n_antenna_dl: N_ANTENNA_DL, /* number of DL antennas */
    n_antenna_ul: N_ANTENNA_UL, /* number of UL antennas */

#if TDD == 1
    uldl_config: 2, /* TDD only */
    sp_config: 7, /* TDD only */
#endif

    n_rb_dl: N_RB_DL, /* Bandwidth: 25: 5 MHz, 50: 10 MHz, 75: 15 MHz, 100: 20 MHz
*/
    cyclic_prefix: "normal",

    phich_duration: "normal",
    phich_resource: "1", /* ratio of NG */

    /* SIB1 */
    si_value_tag: 0, /* increment modulo 32 if SI is modified */
    cell_barred: false, /* SIB1.cellBarred-r13 */
    intra_freq_reselection: true, /* SIB1.intraFreqReselection */
    q_rx_lev_min: -70, /* SIB1.q-RxLevMin */
    p_max: 10, /* maximum power allowed for the UE (dBm) */
    si_window_length: 40, /* ms */
    sib_sched_list: [
     {
       filename: "sib2_3.asn",
       si_periodicity: 16, /* frames */
```

```
      },
    ],

#if N_RB_DL == 6
    si_coderate: 0.30,   /* maximum code rate for SI/RA/P-RNTI messages */
#else
    si_coderate: 0.20,   /* maximum code rate for SI/RA/P-RNTI messages */
#endif
    si_pdcch_format: 2, /* 2 or 3. Log2 of the number of CCEs for PDCCH
                           for SI/RA/P-RNTI */

    n_symb_cch: 0, /* number of symbols for CCH (0 = auto) */

    /* PDSCH dedicated config (currently same for all UEs) */
    pdsch_dedicated: {
#if N_ANTENNA_DL == 4
      p_a: -6,
#elif N_ANTENNA_DL == 2
      p_a: -3,
#else
      p_a: 0,
#endif
      p_b: -1, /* -1 means automatic */
    },

    /* If defined, force for number of CCEs for UE specific PDCCH to
       2^pdcch_format. Otherwise it is computed from the reported
       CQI. Range: 0 to 3. */
#if N_RB_DL == 6
    pdcch_format: 1,
#else
    pdcch_format: 2,
#endif

    /* if defined, force the PDSCH MCS for all UEs. Otherwise it is
       computed from the reported CQI */
    /* pdsch_mcs: 12, */

#if N_RB_DL == 6
    prach_config_index: 15, /* subframe 9 every 20 ms */
#else
    prach_config_index: 4, /* subframe 4 every 10 ms */
#endif
    prach_freq_offset: -1, /* -1 means automatic */

    /* PUCCH dedicated config (currently same for all UEs) */
    pucch_dedicated: {
      n1_pucch_sr_count: 11, /* increase if more UEs are needed */
      cqi_pucch_n_rb: 1, /* increase if more UEs are needed */
#if TDD == 1
```

```
      //tdd_ack_nack_feedback_mode: "bundling", /* TDD only */
      tdd_ack_nack_feedback_mode: "multiplexing", /* TDD only */
#endif
    },

    /* PUSCH dedicated config (currently same for all UEs) */
    pusch_dedicated: {
      beta_offset_ack_index: 9,
      beta_offset_ri_index: 6,
      beta_offset_cqi_index: 6,
    },

    pusch_hopping_offset: -1, /* -1 means automatic */

    /* MCS for Msg3 (=CCCH RRC Connection Request) */
    pusch_msg3_mcs: 0,

    /* this CQI value is assumed when none is received from the UE */
#if N_RB_DL == 6
    initial_cqi: 5,
#else
    initial_cqi: 3,
#endif

    /* if defined, force the PUSCH MCS for all UEs. Otherwise it is
       computed from the last received SRS/PUSCH. */
    //  pusch_mcs: 18,

    dl_256qam: true,
    ul_64qam: true,

    /* Scheduling request period (ms). Must be >= 40 for HD-FDD */
    sr_period: 20,

    /* CQI report config */
    cqi_period: 40, /* period (ms). Must be >= 32 for HD-FDD */

#if N_ANTENNA_DL >= 2
    /* RI reporting is done with a period of m_ri * cqi_period.
       m_ri = 0 (default) disables RI reporting. */
    m_ri: 8,
    /* transmission mode */
    transmission_mode: 3,
#endif

    /* SRS dedicated config. All UEs share these
       parameters. srs_config_index and freq_domain_position are
       allocated for each UE) */
    srs_dedicated: {
#if N_RB_DL == 6
```

```
      srs_bandwidth_config: 7,
      srs_bandwidth: 1,
#elif N_RB_DL == 15
      srs_bandwidth_config: 6,
      srs_bandwidth: 1,
#elif N_RB_DL == 25
      srs_bandwidth_config: 3,
      srs_bandwidth: 1,
#elif N_RB_DL == 50
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#elif N_RB_DL == 75
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#else
      srs_bandwidth_config: 2,
      srs_bandwidth: 3,
#endif
      srs_subframe_config: 3, /* 0 - 15 */
      srs_period: 40, /* period (ms). Must be >= 40 for HD-FDD */
      srs_hopping_bandwidth: 0,
    },

    /* MAC configuration (same for all UEs) */
    mac_config: {
      ul_max_harq_tx: 5, /* max number of HARQ transmissions for uplink */
      dl_max_harq_tx: 5, /* max number of HARQ transmissions for downlink */
    },

    /* CPU load limitation */
    pusch_max_its: 6, /* max number of turbo decoder iterations */

    /* dynamic power control */
    dpc: true,
    dpc_pusch_snr_target: 25,
    dpc_pucch_snr_target: 20,

    /* RRC/UP ciphering algorithm preference. EEA0 is always the last. */
    cipher_algo_pref: [],
    /* RRC integrity algorithm preference. EIA0 is always the last. */
    integ_algo_pref: [2, 1],

    /* (in ms) send RRC connection release after this time of network
       inactivity */
    inactivity_timer: 10000,

    /* SRB configuration */
    srb_config: [
     {
       id: 1,
```

```
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      },
      {
        id: 2 ,
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      }
    ],

    /* DRB configuration */
    drb_config: "drb.cfg",
  },
}
```

Scenario B – Two cells 2x2 MIMO FDD mode

++

./enb/config/enb-scenario-b.cfg

++

```
/* lteenb configuration file version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */

#define N_CELL            2 // should not be changed, used in rf_driver/config.cfg
#define TDD               0 // Values: 0 (FDD), 1(TDD)
#define N_RB_DL           25 // Values: 6 (1.4 MHz), 15 (3MHz), 25 (5MHz), 50 (10MHz), 75 (15MHz), 100 (20MHz)
#define N_ANTENNA_DL      2 // Values: 1 (SISO), 2 (MIMO 2x2)
#define N_ANTENNA_UL      2 // Values: 1, 2
#define CHANNEL_SIM       0 // Values: 0 (channel simulator disabled), 1 (channel simulator enabled)
#define CQI_CONFIG        1 // Values: 0 (periodic CQI), 1 (aperiodic CQI)

{
  /* Log filter: syntax: layer.field=value[,...]

     Possible layers are phy, mac, rlc, pdcp, rrc, nas, s1ap, x2ap, gtpu and
     all. The 'all' layer is used to address all the layers at the
     same time.

     field values:

     - 'level': the log level of each layer can be set to 'none',
     'error', 'info' or 'debug'. Use 'debug' to log all the messages.

     - 'max_size': set the maximum size of the hex dump. 0 means no
       hex dump. -1 means no limit.
  */
  //log_options: "all.level=debug,all.max_size=32",
  log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,s1ap.level=debug,s1ap.max_size=1,x2ap.level=debug,x2ap.max_size=1,rrc.level=debug,rrc.max_size=1",
  log_filename: "/tmp/enb0.log",

  /* Enable remote API and Web interface */
  com_addr: "0.0.0.0:9001",

  /* RF driver configuration */
  include "rf_driver/config.cfg",
```

```
#if CHANNEL_SIM == 1
  rf_ports: [
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    },
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    }
  ],
#endif

  mme_list: [
    {
      /* address of MME for S1AP connection. Must be modified if the MME
         runs on a different host. */
      mme_addr: "127.0.1.100",
    },
  ],
  /* GTP bind address (=address of the ethernet interface connected to
     the MME). Must be modified if the MME runs on a different host. */
  gtp_addr: "127.0.1.1",

  /* high 20 bits of SIB1.cellIdentifier */
  enb_id: 0x1A2D0,

  /* list of cells */
  cell_list: [
    {
      rf_port: 0,
      cell_id: 0x01, /* low 8 bits of SIB1.cellIdentifier */
      tac: 0x0001, /* SIB1.trackingAreaCode */
      n_id_cell: 1,
      root_sequence_index: 204, /* PRACH root sequence index */

      /* carrier aggregation configuration (for rel 10 UEs) */
#if TDD == 1
     dl_earfcn: 40620, /* DL center frequency: 2593 MHz (band 41) */
#else
     dl_earfcn: 3350,  /* DL center frequency: 2680 MHz (Band 7) */
#endif

      /* list of secondary available cells */
      scell_list: [
        {
```

```
          cell_id: 0x02,
          ul_allowed: true,
          cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
        },
      ],
    },
    {
      rf_port: 1,
      cell_id: 0x02, /* low 8 bits of SIB1.cellIdentifier */
      tac: 0x0001, /* SIB1.trackingAreaCode */
      n_id_cell: 2,
      root_sequence_index: 28, /* PRACH root sequence index */
      cell_barred: true,

#if TDD == 1
    dl_earfcn: 39150, /* DL center frequency: 2350 MHz (band 40) */
#else
    dl_earfcn: 1575,  /* DL center frequency: 1842.5 MHz (Band 3) */
#endif

      /* list of secondary available cells */
      scell_list: [
        {
          cell_id: 0x01,
          ul_allowed: true,
          cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
        },
      ],
    },
  ], /* cell_list */

  /* default cell parameters */
  cell_default: {

    /* Broadcasted PLMN identities */
    plmn_list: [
      "00101",
    ],

    n_antenna_dl: N_ANTENNA_DL, /* number of DL antennas */
    n_antenna_ul: N_ANTENNA_UL, /* number of UL antennas */

#if TDD == 1
    uldl_config: 2, /* TDD only */
    sp_config: 7, /* TDD only */
#endif
```

```
    n_rb_dl: N_RB_DL, /* Bandwidth: 25: 5 MHz, 50: 10 MHz, 75: 15 MHz, 100: 20 MHz
*/
    cyclic_prefix: "normal",

    phich_duration: "normal",
    phich_resource: "1", /* ratio of NG */

    /* SIB1 */
    si_value_tag: 0, /* increment modulo 32 if SI is modified */
    cell_barred: false, /* SIB1.cellBarred-r13 */
    intra_freq_reselection: true, /* SIB1.intraFreqReselection */
    q_rx_lev_min: -70, /* SIB1.q-RxLevMin */
    p_max: 10, /* maximum power allowed for the UE (dBm) */
    si_window_length: 40, /* ms */
    sib_sched_list: [
     {
       filename: "sib2_3.asn",
       si_periodicity: 16, /* frames */
     },
    ],

#if N_RB_DL == 6
    si_coderate: 0.30,   /* maximum code rate for SI/RA/P-RNTI messages */
#else
    si_coderate: 0.20,   /* maximum code rate for SI/RA/P-RNTI messages */
#endif
    si_pdcch_format: 2, /* 2 or 3. Log2 of the number of CCEs for PDCCH
                           for SI/RA/P-RNTI */

    n_symb_cch: 0, /* number of symbols for CCH (0 = auto) */

    /* PDSCH dedicated config (currently same for all UEs) */
    pdsch_dedicated: {
#if N_ANTENNA_DL == 4
      p_a: -6,
#elif N_ANTENNA_DL == 2
      p_a: -3,
#else
      p_a: 0,
#endif
      p_b: -1, /* -1 means automatic */
    },

    /* If defined, force for number of CCEs for UE specific PDCCH to
       2^pdcch_format. Otherwise it is computed from the reported
       CQI. Range: 0 to 3. */
#if N_RB_DL == 6
    pdcch_format: 1,
#else
```

```
    pdcch_format: 2,
#endif

    /* if defined, force the PDSCH MCS for all UEs. Otherwise it is
       computed from the reported CQI */
    /* pdsch_mcs: 12, */

#if N_RB_DL == 6
    prach_config_index: 15, /* subframe 9 every 20 ms */
#else
    prach_config_index: 4, /* subframe 4 every 10 ms */
#endif
    prach_freq_offset: -1, /* -1 means automatic */

    simultaneousAckNackAndCQI_format3: true,
    /* PUCCH dedicated config (currently same for all UEs) */
    pucch_dedicated: {
      n1_pucch_sr_count: 11, /* increase if more UEs are needed */
#if CQI_CONFIG == 0
      cqi_pucch_n_rb: 1, /* increase if more UEs are needed */
#else
      cqi_pucch_n_rb: 0,
#endif

      /* number of PUCCH 1b CS resources. It determines
         the maximum number of UEs that can be scheduled in one TTI
         using carrier aggregation with PUCCH 1b CS ack/nack feedback. */
      n1_pucch_an_cs_count: 8,

      /* number of resource blocks for PUCCH 3. It determines
         the maximum number of UEs that can be scheduled in one TTI
         using carrier aggregation with PUCCH 3 ack/nack feedback. */
      n3_pucch_an_n_rb: 0,

#if TDD == 1
      /* TDD ack/nack feedback mode when a rel 10 UE is detected. It
         can be "bundling", "multiplexing", "cs" or "pucch3". By
         default is it the same as tdd_ack_nack_feedback_mode.
      */
      tdd_ack_nack_feedback_mode_r10: "cs",

      //tdd_ack_nack_feedback_mode: "bundling", /* TDD only */
      tdd_ack_nack_feedback_mode: "multiplexing", /* TDD only */
#endif
       /* ack/nack feedback mode when carrier aggregation is
          enabled. It can be "cs" (for at most two scells) or "pucch3"
          (used in all cases if more than two cells).  */
       ack_nack_feedback_mode_ca: "cs",
    },
```

```
    /* PUSCH dedicated config (currently same for all UEs) */
    pusch_dedicated: {
      beta_offset_ack_index: 9,
      beta_offset_ri_index: 6,
      beta_offset_cqi_index: 6,
    },

    pusch_hopping_offset: -1, /* -1 means automatic */

    /* MCS for Msg3 (=CCCH RRC Connection Request) */
    pusch_msg3_mcs: 0,

    /* this CQI value is assumed when none is received from the UE */
#if N_RB_DL == 6
    initial_cqi: 5,
#else
    initial_cqi: 3,
#endif

    /* if defined, force the PUSCH MCS for all UEs. Otherwise it is
       computed from the last received SRS/PUSCH. */
    //  pusch_mcs: 18,

    dl_256qam: true,
    ul_64qam: true,

    /* Scheduling request period (ms). Must be >= 40 for HD-FDD */
    sr_period: 20,

    /* CQI report config */
#if CQI_CONFIG == 0
    cqi_period: 40, /* period (ms). Must be >= 32 for HD-FDD */
#else
    ap_cqi_period: 40,
    ap_cqi_rm: "rm20",
#endif

#if N_ANTENNA_DL >= 2
#if CQI_CONFIG == 0
    /* RI reporting is done with a period of m_ri * cqi_period.
       m_ri = 0 (default) disables RI reporting. */
    m_ri: 8,
#endif
    /* transmission mode */
    transmission_mode: 3,
#endif

    /* SRS dedicated config. All UEs share these
       parameters. srs_config_index and freq_domain_position are
       allocated for each UE) */
```

```
    srs_dedicated: {
#if N_RB_DL == 6
      srs_bandwidth_config: 7,
      srs_bandwidth: 1,
#elif N_RB_DL == 15
      srs_bandwidth_config: 6,
      srs_bandwidth: 1,
#elif N_RB_DL == 25
      srs_bandwidth_config: 3,
      srs_bandwidth: 1,
#elif N_RB_DL == 50
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#elif N_RB_DL == 75
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#else
      srs_bandwidth_config: 2,
      srs_bandwidth: 3,
#endif
      srs_subframe_config: 3, /* 0 - 15 */
      srs_period: 40, /* period (ms). Must be >= 40 for HD-FDD */
      srs_hopping_bandwidth: 0,
    },

    /* MAC configuration (same for all UEs) */
    mac_config: {
      ul_max_harq_tx: 5, /* max number of HARQ transmissions for uplink */
      dl_max_harq_tx: 5, /* max number of HARQ transmissions for downlink */
    },

    /* CPU load limitation */
    pusch_max_its: 6, /* max number of turbo decoder iterations */

    /* dynamic power control */
    dpc: true,
    dpc_pusch_snr_target: 25,
    dpc_pucch_snr_target: 20,

    /* RRC/UP ciphering algorithm preference. EEA0 is always the last. */
    cipher_algo_pref: [],
    /* RRC integrity algorithm preference. EIA0 is always the last. */
    integ_algo_pref: [2, 1],

    /* (in ms) send RRC connection release after this time of network
       inactivity */
    inactivity_timer: 60000,

    /* SRB configuration */
    srb_config: [
```

```
      {
        id: 1,
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      },
      {
        id: 2 ,
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      }
    ],

    /* DRB configuration */
    drb_config: "drb.cfg",
  },
}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

./mme/config/ue_db-ims.cfg

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ue_db: [{
    sim_algo: "xor", /* USIM authentication algorithm: xor, milenage or tuak */
    imsi: "001010123456789", /* Anritsu Test USIM */
//    imsi: "001012345678901", /* Agilent or R&S Test USIM */
    amf: 0x9001, /* Authentication Management Field */
    sqn: "000000000000", /* Sequence Number */
    K: "00112233445566778899aabbccddeeff", /* Anritsu Test USIM */
//    K: "4147494C454E5420544543484E4F0000", /* Agilent Test USIM */
//    K: "000102030405060708090A0B0C0D0E0F", /* R&S Test USIM */

    impi: "001010123456789@ims.mnc001.mcc001.3gppnetwork.org",
    impu: [
       "001010123456789",
       {impu: "tel:0601", imei:"354120803079480"},
    ],
    domain: "ims.mnc001.mcc001.3gppnetwork.org",
    multi_sim: true, /* Experimental */

    /* For standard SIP client */
    /*pwd:  "amarisoft",
    authent_type: "MD5",*/
}, {
    sim_algo: "milenage",
    imsi: "001010000000001",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
```

```
    K: "00112233445566778899AABBCCDDEEFF",
    impu: ["001010000000001", "tel:0600000001"],
    impi: "001010000000001@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000002",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: ["001010000000002", "tel:0600000002"],
    impi: "001010000000002@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000003",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: ["001010000000003", "tel:0600000003"],
    impi: "001010000000003@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000004",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000004",
    impi: "001010000000004@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000005",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000005",
    impi: "001010000000005@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000006",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000006",
    impi: "001010000000006@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
```

```
    imsi: "001010000000007",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000007",
    impi: "001010000000007@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000008",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000008",
    impi: "001010000000008@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000009",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000009",
    impi: "001010000000009@ims.mnc001.mcc001.3gppnetwork.org",
}, {
    sim_algo: "milenage",
    imsi: "001010000000010",
    opc: "000102030405060708090A0B0C0D0E0F",
    amf: 0x9001,
    sqn: "000000000000",
    K: "00112233445566778899AABBCCDDEEFF",
    impu: "001010000000010",
    impi: "001010000000010@ims.mnc001.mcc001.3gppnetwork.org",
}]
```

Scenario C – Three cells 2x2 MIMO FDD mode

++

./enb/config/enb-scenario-c.cfg

++

```
/* lteenb configuration file version 2022-09-16
 * Copyright (C) 2015-2022 Amarisoft
 */

#define N_CELL            3 // should not be changed, used in rf_driver/config.cfg
#define TDD               0 // Values: 0 (FDD), 1(TDD)
#define N_RB_DL           25 // Values: 6 (1.4 MHz), 15 (3MHz), 25 (5MHz), 50 (10MHz), 75 (15MHz), 100 (20MHz)
#define N_ANTENNA_DL      2 // Values: 1 (SISO), 2 (MIMO 2x2)
#define N_ANTENNA_UL      2 // Values: 1, 2
#define CHANNEL_SIM       0 // Values: 0 (channel simulator disabled), 1 (channel simulator enabled)
#define CQI_CONFIG        1 // Values: 0 (periodic CQI), 1 (aperiodic CQI)

{
  /* Log filter: syntax: layer.field=value[,...]

     Possible layers are phy, mac, rlc, pdcp, rrc, nas, s1ap, x2ap, gtpu and
     all. The 'all' layer is used to address all the layers at the
     same time.

     field values:

     - 'level': the log level of each layer can be set to 'none',
     'error', 'info' or 'debug'. Use 'debug' to log all the messages.

     - 'max_size': set the maximum size of the hex dump. 0 means no
       hex dump. -1 means no limit.
  */
  //log_options: "all.level=debug,all.max_size=32",
  log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,s1ap.level=debug,s1ap.max_size=1,x2ap.level=debug,x2ap.max_size=1,rrc.level=debug,rrc.max_size=1",
  log_filename: "/tmp/enb0.log",

  /* Enable remote API and Web interface */
  com_addr: "0.0.0.0:9001",

  /* RF driver configuration */
  include "rf_driver/config.cfg",
```

```
#if CHANNEL_SIM == 1
  rf_ports: [
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    },
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    },
    {
      channel_dl: {
        type: "awgn",
        noise_level: -30,
      },
    }
  ],
#endif

  mme_list: [
    {
      /* address of MME for S1AP connection. Must be modified if the MME
         runs on a different host. */
      mme_addr: "127.0.1.100",
    },
  ],
  /* GTP bind address (=address of the ethernet interface connected to
     the MME). Must be modified if the MME runs on a different host. */
  gtp_addr: "127.0.1.1",

  /* high 20 bits of SIB1.cellIdentifier */
  enb_id: 0x1A2D0,

  /* list of cells */
  cell_list: [
    {
      rf_port: 0,
      cell_id: 0x01, /* low 8 bits of SIB1.cellIdentifier */
      tac: 0x0001, /* SIB1.trackingAreaCode */
      n_id_cell: 1,
      root_sequence_index: 204, /* PRACH root sequence index */

      /* carrier aggregation configuration (for rel 10 UEs) */
#if TDD == 1
     dl_earfcn: 40620, /* DL center frequency: 2593 MHz (band 41) */
#else
```

```
    dl_earfcn: 3350,  /* DL center frequency: 2680 MHz (Band 7) */
#endif

    /* list of secondary available cells */
    scell_list: [
      {
        cell_id: 0x02,
        ul_allowed: true,
        cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
      },
      {
        cell_id: 0x03,
        ul_allowed: true,
        cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
      },
    ],
  },
  {
      rf_port: 1,
      cell_id: 0x02, /* low 8 bits of SIB1.cellIdentifier */
      tac: 0x0001, /* SIB1.trackingAreaCode */
      n_id_cell: 2,
      root_sequence_index: 28, /* PRACH root sequence index */
      cell_barred: true,

#if TDD == 1
    dl_earfcn: 39150, /* DL center frequency: 2350 MHz (band 40) */
#else
    dl_earfcn: 1575,  /* DL center frequency: 1842.5 MHz (Band 3) */
#endif

    /* list of secondary available cells */
    scell_list: [
      {
        cell_id: 0x01,
        ul_allowed: true,
        cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
      },
      {
        cell_id: 0x03,
        ul_allowed: true,
        cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
```

```
      },
    ],
  },
  {
    rf_port: 2,
    n_id_cell: 3,
    cell_id: 0x03,
    tac: 0x0001,
    root_sequence_index: 202, /* PRACH root sequence index */
    cell_barred: true,

#if TDD == 1
    dl_earfcn: 46090, /* DL center frequency: 753.0 (band 44) */
#else
    dl_earfcn: 6300,  /* DL center frequency: 806.0 MHz (Band 20) */
#endif

    /* list of secondary available cells */
    scell_list: [
      {
        cell_id: 0x01,
        ul_allowed: true,
        cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
      },
      {
        cell_id: 0x02,
        ul_allowed: true,
        cross_carrier_scheduling: false,
//          cross_carrier_scheduling: true,
//          scheduling_cell_id: 0x01,
      }
    ],
  }
], /* cell_list */

  /* default cell parameters */
  cell_default: {

    /* Broadcasted PLMN identities */
    plmn_list: [
      "00101",
    ],

    n_antenna_dl: N_ANTENNA_DL, /* number of DL antennas */
    n_antenna_ul: N_ANTENNA_UL, /* number of UL antennas */

#if TDD == 1
    uldl_config: 2, /* TDD only */
```

```
    sp_config: 7, /* TDD only */
#endif

    n_rb_dl: N_RB_DL, /* Bandwidth: 25: 5 MHz, 50: 10 MHz, 75: 15 MHz, 100: 20 MHz
*/
    cyclic_prefix: "normal",

    phich_duration: "normal",
    phich_resource: "1", /* ratio of NG */

    /* SIB1 */
    si_value_tag: 0, /* increment modulo 32 if SI is modified */
    cell_barred: false, /* SIB1.cellBarred-r13 */
    intra_freq_reselection: true, /* SIB1.intraFreqReselection */
    q_rx_lev_min: -70, /* SIB1.q-RxLevMin */
    p_max: 10, /* maximum power allowed for the UE (dBm) */
    si_window_length: 40, /* ms */
    sib_sched_list: [
     {
      filename: "sib2_3.asn",
      si_periodicity: 16, /* frames */
     },
    ],

#if N_RB_DL == 6
    si_coderate: 0.30,   /* maximum code rate for SI/RA/P-RNTI messages */
#else
    si_coderate: 0.20,   /* maximum code rate for SI/RA/P-RNTI messages */
#endif
    si_pdcch_format: 2, /* 2 or 3. Log2 of the number of CCEs for PDCCH
                           for SI/RA/P-RNTI */

    n_symb_cch: 0, /* number of symbols for CCH (0 = auto) */

    /* PDSCH dedicated config (currently same for all UEs) */
    pdsch_dedicated: {
#if N_ANTENNA_DL == 4
      p_a: -6,
#elif N_ANTENNA_DL == 2
      p_a: -3,
#else
      p_a: 0,
#endif
      p_b: -1, /* -1 means automatic */
    },

    /* If defined, force for number of CCEs for UE specific PDCCH to
       2^pdcch_format. Otherwise it is computed from the reported
       CQI. Range: 0 to 3. */
#if N_RB_DL == 6
```

```
    pdcch_format: 1,
#else
    pdcch_format: 2,
#endif

    /* if defined, force the PDSCH MCS for all UEs. Otherwise it is
       computed from the reported CQI */
    /* pdsch_mcs: 12, */

#if N_RB_DL == 6
    prach_config_index: 15, /* subframe 9 every 20 ms */
#else
    prach_config_index: 4, /* subframe 4 every 10 ms */
#endif
    prach_freq_offset: -1, /* -1 means automatic */

    simultaneousAckNackAndCQI_format3: true,
    /* PUCCH dedicated config (currently same for all UEs) */
    pucch_dedicated: {
      n1_pucch_sr_count: 11, /* increase if more UEs are needed */
#if CQI_CONFIG == 0
      cqi_pucch_n_rb: 1, /* increase if more UEs are needed */
#else
      cqi_pucch_n_rb: 0,
#endif

      /* number of PUCCH 1b CS resources. It determines
         the maximum number of UEs that can be scheduled in one TTI
         using carrier aggregation with PUCCH 1b CS ack/nack feedback. */
      n1_pucch_an_cs_count: 0,

#if TDD == 1
      /* TDD ack/nack feedback mode when a rel 10 UE is detected. It
         can be "bundling", "multiplexing", "cs" or "pucch3". By
         default is it the same as tdd_ack_nack_feedback_mode.
      */
      tdd_ack_nack_feedback_mode_r10: "cs",

      //tdd_ack_nack_feedback_mode: "bundling", /* TDD only */
      tdd_ack_nack_feedback_mode: "multiplexing", /* TDD only */
#endif
       /* ack/nack feedback mode when carrier aggregation is
          enabled. It can be "cs" (for at most two scells) or "pucch3"
          (used in all cases if more than two cells).  */
       ack_nack_feedback_mode_ca: "pucch3",

       /* number of resource blocks for PUCCH 3. It determines
          the maximum number of UEs that can be scheduled in one TTI
          using carrier aggregation with PUCCH 3 ack/nack feedback. */
       n3_pucch_an_n_rb: 3,
```

```
    },

    /* PUSCH dedicated config (currently same for all UEs) */
    pusch_dedicated: {
      beta_offset_ack_index: 9,
      beta_offset_ri_index: 6,
      beta_offset_cqi_index: 6,
    },

    pusch_hopping_offset: -1, /* -1 means automatic */

    /* MCS for Msg3 (=CCCH RRC Connection Request) */
    pusch_msg3_mcs: 0,

    /* this CQI value is assumed when none is received from the UE */
#if N_RB_DL == 6
    initial_cqi: 5,
#else
    initial_cqi: 3,
#endif

    /* if defined, force the PUSCH MCS for all UEs. Otherwise it is
       computed from the last received SRS/PUSCH. */
    //  pusch_mcs: 18,

    dl_256qam: true,
    ul_64qam: true,

    /* Scheduling request period (ms). Must be >= 40 for HD-FDD */
    sr_period: 20,

    /* CQI report config */
#if CQI_CONFIG == 0
    cqi_period: 40, /* period (ms). Must be >= 32 for HD-FDD */
#else
    ap_cqi_period: 40,
    ap_cqi_rm: "rm20",
#endif

#if N_ANTENNA_DL >= 2
#if CQI_CONFIG == 0
    /* RI reporting is done with a period of m_ri * cqi_period.
       m_ri = 0 (default) disables RI reporting. */
    m_ri: 8,
#endif
    /* transmission mode */
    transmission_mode: 3,
#endif

    /* SRS dedicated config. All UEs share these
```

```
       parameters. srs_config_index and freq_domain_position are
       allocated for each UE) */
    srs_dedicated: {
#if N_RB_DL == 6
      srs_bandwidth_config: 7,
      srs_bandwidth: 1,
#elif N_RB_DL == 15
      srs_bandwidth_config: 6,
      srs_bandwidth: 1,
#elif N_RB_DL == 25
      srs_bandwidth_config: 3,
      srs_bandwidth: 1,
#elif N_RB_DL == 50
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#elif N_RB_DL == 75
      srs_bandwidth_config: 2,
      srs_bandwidth: 2,
#else
      srs_bandwidth_config: 2,
      srs_bandwidth: 3,
#endif
      srs_subframe_config: 3, /* 0 - 15 */
      srs_period: 40, /* period (ms). Must be >= 40 for HD-FDD */
      srs_hopping_bandwidth: 0,
    },

    /* MAC configuration (same for all UEs) */
    mac_config: {
      ul_max_harq_tx: 5, /* max number of HARQ transmissions for uplink */
      dl_max_harq_tx: 5, /* max number of HARQ transmissions for downlink */
    },

    /* CPU load limitation */
    pusch_max_its: 6, /* max number of turbo decoder iterations */

    /* dynamic power control */
    dpc: true,
    dpc_pusch_snr_target: 25,
    dpc_pucch_snr_target: 20,

    /* RRC/UP ciphering algorithm preference. EEA0 is always the last. */
    cipher_algo_pref: [],
    /* RRC integrity algorithm preference. EIA0 is always the last. */
    integ_algo_pref: [2, 1],

    /* (in ms) send RRC connection release after this time of network
       inactivity */
    inactivity_timer: 10000,
    /* SRB configuration */
```

```
    srb_config: [
      {
        id: 1,
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      },
      {
        id: 2 ,
        maxRetxThreshold: 32,
        t_Reordering: 45,
        t_PollRetransmit: 60,
      }
    ],
    /* DRB configuration */
    drb_config: "drb.cfg",
  },
}
```

Figures Bibliography

Figure 2.1.1: Architecture of the Evolved Packet Network.

“EPC and 4G Packet Networks – Driving the Mobile Broadband Revolution” of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan

Figure 2.1.2: EUTRA Operating Bands.

“EPC and 4G Packet Networks – Driving the Mobile Broadband Revolution” of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan

Figure 2.1.3: Categories of LTE devices.

“EPC and 4G Packet Networks – Driving the Mobile Broadband Revolution” of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan

Figure 2.1.4: LTE protocol stack.

https://www.prodevelopertutorial.com/lte-chapter-11-lte-user-plane-protocol-stack/

Figure 2.2.1: Benefits of Open RAN.

https://www.juniper.net/it/it/research-topics/what-is-open-ran.html

Figure 2.2.2: Components of disaggregated RAN.

https://www.mavenir.com/portfolio/mavair/radio-access/openran/

Figure 2.2.3: Moving network functionalities towards the cloud.

https://www.nokia.com/about-us/newsroom/articles/open-ran-explained/

Figure 2.2.4: O-RAN architecture.

https://www.o-ran.org/

Figure 3.1.1: Amarisoft SDR card.

https://www.amarisoft.com/products/custom-projects/

Figure 3.4.1: AMARI Callbox architecture.

https://tech-academy.amarisoft.com/ConfigurationGuide.html

Figure 3.4.2: QCIs and their services.

https://www.sharetechnote.com/html/Handbook_LTE_QCI.html

Figure 3.4.3: Connections between LTEMME and LTEIMS.

https://tech-academy.amarisoft.com/appnote_ims.doc

Figure 4.2.1: Theoretical attachment procedure.

“EPC and 4G Packet Networks – Driving the Mobile Broadband Revolution” of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan

Figure 4.4.1: VoLTE Architecture.

“EPC and 4G Packet Networks – Driving the Mobile Broadband Revolution” of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan

Figure 4.5.1: Example two cells in the same SDR.

https://tech-academy.amarisoft.com/appnote_handover.doc

Figure 5.1.1: UL physical layer parameter values set by ue-Category.

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2434

Figure 5.1.2: DL physical layer parameter values set by ue-Category.

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2434

Figure 5.1.3: DL physical layer parameter values set by ue-CategoryDL – Part 1.

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2434

Figure 5.1.4: DL physical layer parameter values set by ue-CategoryDL – Part 2.

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2434

Figure 5.1.5: UL physical layer parameter values set by ue-CategoryUL.

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2434

Bibliography

“EPC and 4G Packet Networks – Driving the Mobile Broadband Revolution” of M. Olsson, S. Sultana, S. Rommer, L. Frid and C. Mulligan

“IMS Application Developer’s Handbook – Creating and Deploying Innovative IMS Applications” of R. Noldus, U. Olsson, C. Mulligan, I. Fikouras, A. Ryde and M. Stille

“Open Radio Access Network (O-RAN) Systems Architecture and Design” of Wim Rouwet

https://tech-academy.amarisoft.com/

https://www.mpirical.com/glossary/udc-unified-data-convergence

https://www.sqimway.com/nr_band.php

https://www.sqimway.com/nr_raster.php

https://sdn.ieee.org/newsletter/december-2017/network-slicing-and-3gpp-service-and-systems-aspects-sa-standard

https://www.3gpp.org/specifications-technologies

http://lte-epc.blogspot.com/2009/12/lte-bearer-service-architecture.html

http://www.sharetechnote.com/html/Handbook_LTE_PDSCH.html

https://www.sharetechnote.com/html/Handbook_LTE_ServCellID_SCellID.html

https://info-nrlte.com/tag/cross-carrier-scheduling/

http://www.sharetechnote.com/html/Handbook_LTE_DownlinkPowerAllocation.html

https://www.mpirical.com/glossary/dnn-data-network-name

http://www.sharetechnote.com/html/FrameStructure_DL.html#PhysicalChannels_and_Signals_in_RadioFrame

https://www.techplayon.com/5g-ran-and-5gc-network-slice-signaling/

https://www.sharetechnote.com/html/Handbook_LTE_PUCCH_Format1_Location.html

https://patents.google.com/patent/US20190357116A1/en#:~:text=The%20Forbidden%20PLMN%20list%20is%20used%20to%20avoid,try%20to%20attach%20or%20register%20to%20the%20network.

https://www.sharetechnote.com/html/Handbook_CellSelection_PLMN.html

https://www.techplayon.com/plmn-selection-in-lte-idle-mode-action/

https://www.sharetechnote.com/html/Handbook_LTE_Authentication.html

https://www.emcu-homeautomation.org/identificativo-mcc-mnc-degli-operatori-mobili-italiani/

https://www.sharetechnote.com/html/Handbook_LTE_EEA.html

https://nickvsnetworking.com/hss-usim-authentication-in-lte-nr-4g-5g/

https://nickvsnetworking.com/usim-basics/

https://imei.org/blog/imsi-number

http://smstools3.kekekasvi.com/topic.php?id=288

https://www.sharetechnote.com/html/Handbook_LTE_T3410_T3450.html

http://www.sharetechnote.com/html/OpenRAN/OR_open5gs_webui.html#Making_WebUI_accessible_from_remote_PC

https://www.pslightwave.com/how-to-measure-network-reliability/

https://ieeexplore.ieee.org/document/45203

https://obkio.com/blog/how-to-measure-network-performance-metrics/#the-most-important-network-metrics

https://obkio.com/blog/measuring-voip-quality-with-mos-score-mean-opinion-score/

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2434

https://www.3gpp.org/specifications

https://www.researchgate.net/figure/Three-types-of-CA-in-LTE-A-4_fig1_271451859

https://www.juniper.net/documentation/en_US/junos-mobility11.2/topics/concept/gtp-mobility-protocols-overview.html

https://teletopix.org/4g-lte/what-is-cyclic-prefix-in-lte/#:~:text=What%20is%20Cyclic%20prefix%20in%20LTE%20%3F%20The,of%20the%20last%20part%20of%20the%20following%20symbol.

http://www.sharetechnote.com/html/Handbook_LTE_PHICH_PHICHGroup.html

https://destevez.net/2022/06/lte-downlink-phich/

https://www.rfwireless-world.com/Terminology/LTE-MIB-SIB-system-information-blocks.html

https://www.4gmobiletech.com/lte-protocol-stack-channels

https://www.tutorialspoint.com/lte/lte_protocol_stack_layers.htm

https://www.prodevelopertutorial.com/lte-chapter-12-lte-control-plane-protocol-stack/

https://www.prodevelopertutorial.com/lte-chapter-11-lte-user-plane-protocol-stack/

https://www.rfwireless-world.com/Tutorials/LTE-Protocol-Stack.html#:~:text=The%20user%20plane%20LTE%20protocol%20stack%20consists%20of,NAS%3A%20In%20the%20uplink%20it%20does%20packet%20filtering.

https://telecompedia.net/sib1-in-lte/

https://www.sharetechnote.com/html/Handbook_LTE_ResourceAllocation_ManagementUnit.html

https://www.sharetechnote.com/html/Handbook_LTE_QCI.html

https://www.sharetechnote.com/html/LTE_TDD_Overview.html

https://tech-academy.amarisoft.com/appnote_handover.doc

https://www.juniper.net/it/it/research-topics/what-is-open-ran.html

https://www.mavenir.com/portfolio/mavair/radio-access/openran/

https://www.nokia.com/about-us/newsroom/articles/open-ran-explained/

https://www.ericsson.com/en/openness-innovation/open-ran-explained

https://www.o-ran.org/

https://www.mavenir.com/portfolio/mavair/radio-access/openran/

https://www.telefonica.com/es/sala-comunicacion/telefonica-y-nec-construiran-pilotos-de-open-ran-en-4-mercados-un-hito-clave-hacia-el-despliegue-masivo/

https://www.dell.com/en-us/blog/open-ran-bringing-it-all-together/

https://www.dday.it/redazione/39366/cosa-si-intende-per-open-ran-e-perche-ne-sentiremo-sempre-piu-parlare

https://www.key4biz.it/open-ran-cose-e-perche-e-importante-per-lo-sviluppo-del-5g/358177/

https://www.cisco.com/c/en/us/solutions/what-is-open-ran.html

https://ormuco.com/blog/network-virtualization-how-do-sdn-nfv-differ

https://www.checkpoint.com/cyber-hub/network-security/what-is-network-function-virtualization-nfv/

https://sase.vmware.com/content/dam/digitalmarketing/vmware-sase/pdfs/208805aq-so-vcloud-guide-sd-wan-nfv-vfn-uslet-web.pdf

https://www.micronova.de/en/telco/news/trends-topics/article/open-ran-sdn-and-nfv-for-radio-access-networks.html

https://www.ibm.com/cloud/blog/software-defined-networking

https://www.amarisoft.com/products/network-deployment/

https://www.eeworldonline.com/open-ran-design-options-from-networks-to-memory/

https://www.5g-networks.net/5g-technology/open-ran-ric-open-ran-intelligent-controller/

https://cacombos.com/

https://5g-tools.com/4g-lte-earfcn-calculator/

https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/QualiPoc_Android_bro_en_3607-1607-12_v0400.pdf

https://www.rohde-schwarz.com/in/products/test-and-measurement/network-data-collection/qualipoc-android_63493-55430.html

https://www.keysight.com/us/en/product/NTN00000B/nemo-analyze-drive-test-post-processing-solution.html

https://tele-tools.com/nemo-analyzer/

https://tech-academy.amarisoft.com/appnote_calibration.doc

https://www.fpgakey.com/technology/details/application-of-fpga-in-software-radio#:~:text=SDR%20baseband%20processing%20usually%20requires%20a%20processor%20and,data%20channel%20and%20control%20to%20minimize%20system%20delay.

https://www.amarisoft.com/products/custom-projects/

https://www.fpgakey.com/technology/details/application-of-fpga-in-software-radio

https://www.sharetechnote.com/html/Handbook_LTE_QCI.html

https://en.wikipedia.org/wiki/QoS_Class_Identifier

https://www.keysight.com/us/en/product/NTH00000B/nemo-handy-handheld-measurement-solution.html

https://www.telecomtutorial.info/post/volte-sip-ims-registration-call-flow-procedure-default-vs-dedicated-bearer-in-lte

https://it.wikipedia.org/wiki/Session_Initiation_Protocol

https://en.wikipedia.org/wiki/Multimedia_telephony

https://blog.3g4g.co.uk/2013/08/volte-bearers.html

https://www.tutorialspoint.com/session_initiation_protocol/session_initiation_protocol_sdp.htm

https://edwinhernandez.com/2019/12/27/measurement-reports-in-4g-lte-part-1/

https://www.prodevelopertutorial.com/lte-handover-events-measurement-reports/

https://www.sharetechnote.com/html/Handbook_LTE_MultiCell_Measurement_LTE.html

https://www.sharetechnote.com/html/5G/5G_LTE_Interworking.html

https://devopedia.org/dual-connectivity#:~:text=Dual%20Connectivity%20%28DC%29%20is%20a%20feature%20that%20was,to%20both%20LTE%20E-UTRA%20and%205G%20NR%20nodes.

https://blog.3g4g.co.uk/2020/06/carrier-aggregation-ca-and-dual.html

https://www.bing.com/search?q=measurement+report+in+lte&cvid=af9bddc11cec4bf58938e3f5127026f3&aqs=edge.0.69i64i450l8...8.2970859j0j1&FORM=ANNTA1&PC=U531

https://tech-academy.amarisoft.com/appnote_calibration.doc#6f469671ee0bc8e1248ef94eeeef0d03

https://www.parallelwireless.com/blog/the-future-of-open-ran-7-predictions-for-2022/
https://www.techrepublic.com/article/open-ran-benefits/

https://www.globenewswire.com/en/news-release/2022/03/08/2398558/28124/en/Global-OPEN-RAN-Market-Outlook-An-Opportunity-worth-32-Billion-by-2030.html

https://www.kbvresearch.com/europe-open-radio-access-network-market/

https://the-mobile-network.com/2023/02/heres-the-o-ran-product-news-ahead-of-mwc23/

https://rimedolabs.com/blog/ran-intelligent-controller-ric-overview-xapps-and-rapps/

https://www.rcrwireless.com/20211123/fundamentals/xapps-vs-rapps-network-automation-fundamentals

https://stlpartners.com/articles/telco-cloud/ric-xapps-rapps-who-are-the-key-players/

https://www.whiteboxsolution.com/blog/open-ran-vs-virtual-ran-explained/#:~:text=Open%20RAN%20%28O-RAN%29%20is%20the%20foundation%20for%20building,the%20lack%20of%20an%20interoperable%20and%20flexible%20structure.

https://www.ericsson.com/en/blog/2020/2/virtualized-5g-ran-why-when-and-how

https://www.samsung.com/global/business/networks/insights/press-release/0222-dish-wireless-launches-virtual-open-ran-5g-network-with-samsung/

https://www.whiteboxsolution.com/blog/open-ran-vs-virtual-ran-explained/

https://telco.vmware.com/solutions/ran.html

https://moniem-tech.com/questions/what-is-the-difference-between-open-ranvran-and-o-ran/

https://www.5gtechnologyworld.com/deploy-and-maintain-an-open-ran-network/#:~:text=Mobile%20network%20operators%20%28MNOs%29%20throughout%20the%20world%2C%20including,offer%20new%20ways%20to%20manage%20the%20mobile%20network.

https://strandconsult.dk/blog/open-ran-the-future-not-the-present/

https://www.parallelwireless.com/blog/the-future-of-open-ran-7-predictions-for-2022/