UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Study of RAN slicing using SRS LTE in a real SDN-NFV platform

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Ali Esmaeily

In partial fulfilment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisors: Anna Umbert Juliana

Katerina Koutlia

Barcelona, June 2018

Title of the thesis: Study of RAN slicing using SRS LTE in a real SDN-NFV platform

Author: Ali Esmaeily

Advisors: Anna Umbert Juliana, Katerina Koutlia

Abstract

In today's mobile communication networks tendency for flexibility in management and efficient network operation in one hand, while maintaining the capital and operating expenditures low on the other hand, given the opportunities for proposing new technologies which are able to cope with the continuous increase of traffic demand which probably could congest and overflow current networks. Nowadays virtualization as a key technology enables network operators to have an undeniable progress since it decouples software applications from the underlying hardware. SDN and NFV as the two main parts in network virtualization are promising approaches that when combined can provide the necessary flexibility in the network and resource management. These capabilities make it possible to share the available resources dynamically to slice the Radio Access Network (RAN) according to the specific requests.

To contribute these concepts in this project, a real-time SDN-NFV platform, which splits into OpenAirInterface, srsLTE, and 5G-EmPOWER platforms, has been studied, configured and integrated. Moreover, RAN Slicing has been applied for different Mobile Virtual Network Operator (MVNO) or tenants and also several connectivity tests have been performed throughout the project to confirm the network functionality.

Acknowledgements

I would like to express my gratitude to my advisors, Prof. Anna Umbert Juliana and Dr. Katerina Koutlia, for their support, patience, and encouragement throughout the project. Their technical and editorial advice was essential to the completion of this dissertation and taught me innumerable lessons and insights in the workings on academic projects. My thanks also go to the research engineer of CreateNet in Trento Italy, Mr. Kewin Rausch for his significant help during the project implementation and also to the network administrator of GRCM lab, Mr. Sergio Garcia, for his technical support.

Last, but not least, I would like to thank my family for their understanding and love during my entire life.

Revision history and approval record

Revision	Date	Purpose
0	07/05/2018	Document creation
1	12/06/2018	Document revision

Written by:		Reviewed and approved by:	
Date	07/05/2018	Date	12/06/2018
Name	Ali Esmaeily	Name	Anna Umbert Juliana Katerina Koutlia
Position	Project Author	Position	Project Supervisors

Table of contents

List of Figures

List of Tables

1. Introduction

In the recent years, the demand for exchanging information with appropriate Quality of Service (QoS) through mobile communication networks has had undeniable growth especially due to the fact that Internet, as an infrastructure, facilitates data traffic communication. Regarding to the new developments in the mobile communication standards, such as Long Term Evolution (LTE) and its next releases LTE Advanced (LTE-A) and LTE-A-Pro, this amount of traffic can be easily managed in faster and efficient way with much better QoS than the previous standards, such as Global System for Mobile communications (GSM) and Universal Mobile Telecommunications System (UMTS). However, a higher global traffic demand is expected by 2020, which will reach the 30.6 Exabyte's monthly [1]. This amount of exchanging information will cause traffic congestion in the mobile networks and as a result, will also overflow the LTE network. Therefore, new approaches have been pursued in order to deploy denser and heterogeneous networks (HetNet).

Virtualization is one of the new required approaches that have been followed in the last years, which allows decoupling the software applications from the underlying hardware. Virtualization, in turn, is enabled by two fundamental technologies which are the Software Defined Networking (SDN) and Network Function Virtualization (NFV). By the combination of these two techniques, flexibility in the network and management of the resources can be obtained. Moreover, according to the SDN and NFV capabilities, the possibility of dynamically sharing the radio resources and slicing the Radio Access Network (RAN) based on the requested specifications can be achieved.

RAN slicing [2], as a technique, can cope with the today`s expansion of traffic demands, even if the traffic in the wireless networks follows heterogeneous patterns. One of the hot topics of these days is that according to the data traffic variations in the network, each Mobile Virtual Network Operator (MVNO) or tenant can have different requests in a specific node, such as in a Wi-Fi Access Point or a LTE eNB. As such, RAN slicing as a solution can decrease the expenses in the deployment and operation phase by creating different logical networks over a shared physical network infrastructure in a way that each logical network can be independent of others and can carry out particular applications for each Communications Service Providers (CSP) based on the different needs. Furthermore, each of the CSPs (or tenants) can manage its own slice of radio resources while providing the proper set of services for its users.

In order to study the RAN Slicing concept under real-time scenarios in HetNets there is the need for a platform with SDN/NFV capabilities. Currently, some research groups in mobile communications area are investigating in this field. More specifically at UPC, the Mobile Communications Research Group (GRCM) is collaborating with the Future Networks (FuN) research Unit, which belongs to the Fondazione Bruno Kessler Create-Net (FBK.Create-Net) international research centre in Trento, in the study of the RAN Slicing concept on a real-time test platform, known as 5G-EmPOWER. This platform implements Wi-Fi and LTE networks following the SDN paradigm and providing virtualization functionalities.

The collaboration of GRCM and FuN groups resulted in the implementation of a RAN slicing strategy in WLAN networks. Currently, both groups work together with purpose to

implement RAN Slicing for the LTE standard. This project forms part of this collaboration with target to contribute to the RAN Slicing implementation.

Based on the above, the main objective of this project is to study the functionality and capabilities of the 5G-EmPOWER platform with ultimate goal to be able to perform modifications on the SW. This knowledge will be finally used for the implementation of the RAN Slicing approach. 5G-EmPOWER as the controller section in this project has been designed by the research centre FDK-Create-Net. It has been used for integrating with the Software Radio Systems (SRS) LTE as the RAN section which is provided by Software Radio Systems Limited, a company founded by a team of wireless experts from the CONNECT research centre at Trinity College Dublin. For its part, OpenAirInterface (OAI) LTE, created by the OpenAirInterface Software Alliance (OSA), has been implemented to play the role of Evolved Packet Core (EPC) of LTE network in this project.

This project consists of the following chapters:

- In chapter 2, the content of the project has been defined through the explanation of several principle concepts such as SDN, NFV and LTE standard. Moreover, an introduction of the Real-time SDN-NFV platform used in this project that consists of the OAI LTE (as EPC section), the srsLTE (as RAN section) and last the 5G-EmPOWER (as Controller section), will be given afterwards. Let us notice that these three entities are clarified throughout this project.
- The methodology that has been pursued to carry out this work is briefly discussed in chapter 3.
- Chapter 4 exposes all the procedures developed to reach the main objective and the obtained results.
- Finally, in chapter 5, the conclusions of the thesis and the future work are presented.

The following Gantt chart illustrates the progress procedure of the project:

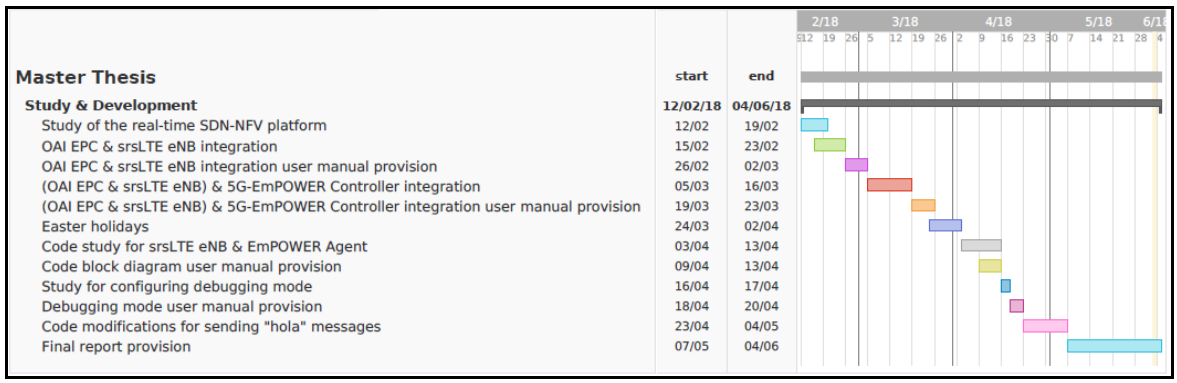


Table 1. Gantt chart of the Master Thesis.

As indicated in Table 1, during the project progress several user manuals have been created. These user manuals are available online at the corresponding links listed in the Annex sections.

2. State of the Art

In this chapter, the context in which the project is carried out is presented. First, different technologies and standards such as SDN, NFV, and LTE which are involved in this project are discussed. Then, the Real-time SDN-NFV platform that is the combination of srsLTE, OpenAirInterface LTE and 5G-EmPOWER platforms is described.

2.1. Software Defined Networking (SDN)

In traditional networks, there is a range of hardware devices; primarily routers, switches, and firewalls. These devices include hardware connectivity, which moves the data through them, and software elements that are configured to control the movement of data through the hardware. Such networks behave like closed systems that have a limited range of control on their nodes. Moreover, according to their characteristics, they do not give the opportunity to for updating the employed network infrastructure, since, on one hand, the existing protocols cannot be modified and on the other hand, there is no feasibility to create new protocols that will adapt the network to the users` demands.

The main obstacle in the way towards evolution of current networks lies on the fact that the user plane and control plane are coupled. This means that each device in the network makes its own forwarding decisions. As such, developing new applications and functionalities in the elements of the network turns to be more intricate, since any modifications in applications and/or functionalities have to be configured on all the involved elements of the infrastructure.

Software Defined Networking (SDN) [3] functionally enables the network to be accessed programmatically, allowing for automated management and orchestration techniques; application of configuration policy across multiple routers, switches, and servers; and the decoupling of the application that performs these operations from the network devices' operating system. In other words, this new paradigm separates the control and data planes, that is, the physical infrastructure from the logic control, by providing the network with a common software control. This decoupling allows direct virtualization which in turn makes the implementation of new protocols and management easy and applicable.

SDN is also an architecture that allows for a logically centralized intelligence and distributed management, where policy that dictates the forwarding rules is centralized, while the actual forwarding rule processing is distributed among multiple devices. In this model, application policy calculation (e.g. QoS, access control lists and tunnel creation) happens locally in real time and the quality, security, and monitoring of policies are managed centrally and then pushed to the switching/routing nodes. This allows for more flexibility, control, and scalability of the network itself, and the use of templates, variables, multiple databases of users, and policies.

Figure 1 depicts the evolution from a current network to a SDN network, in which the forwarding hardware is separated from the control logic. In the SDN network the data-path is represented by solid lines and the control-path is represented by dashed lines.

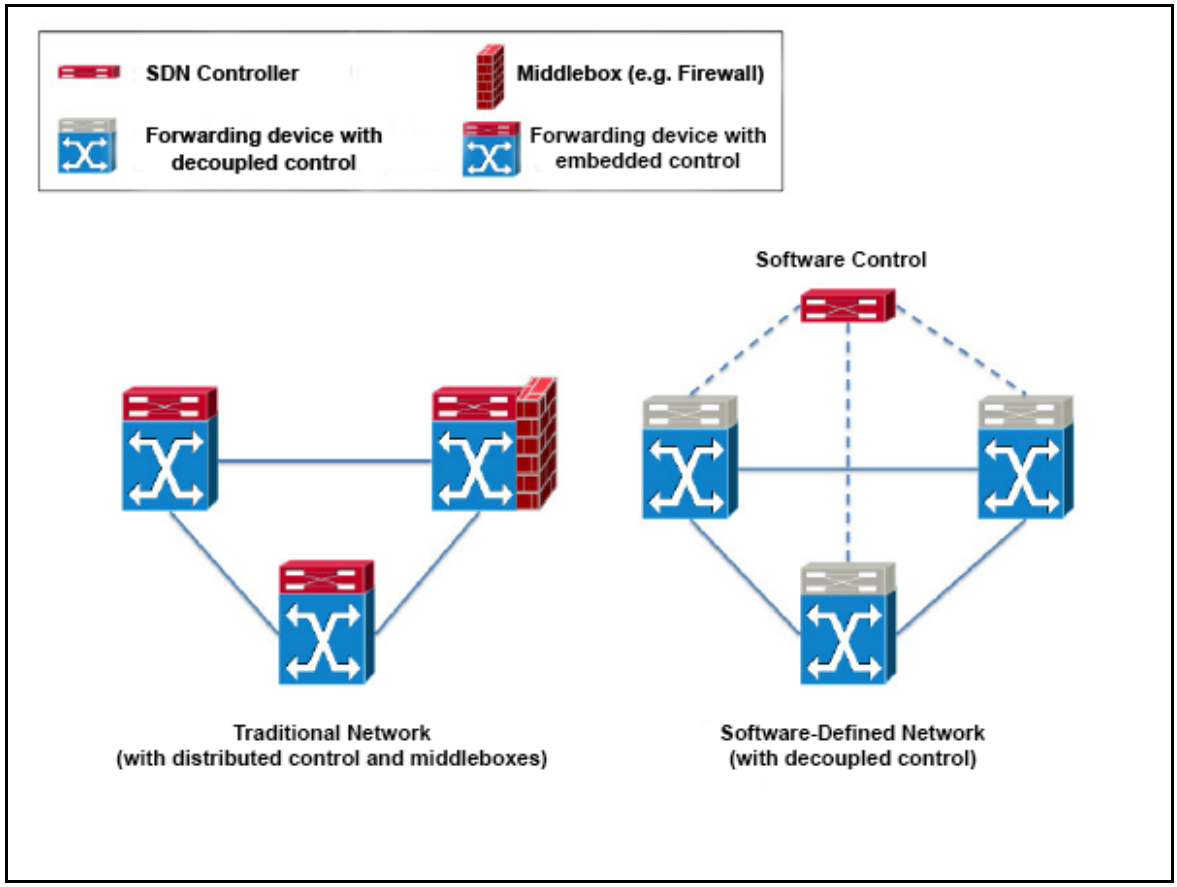


Figure 1. Architecture of a traditional network and a SDN network.

2.1.1. SDN Architecture

The SDN architecture is organized in the following components (see Figure 2):

- **Application layer**

  In this layer, different programs are running and interacting with the controller layer through the Northbound interface in order to keep the controller informed about the network demands.

- **Controller layer**

  The role of this logical entity is to interpret requests from the application layer towards the underlying data plane and to prepare an overview of the network by means of statistics and events to the application layer.

- **Infrastructure layer**

  These logical elements, which in fact are special switches, can be programmed through the controller layer of the network and expose visibility and uncontested control over their advertised forwarding and data processing capabilities. They communicate with the controller layer through the Southbound interface.

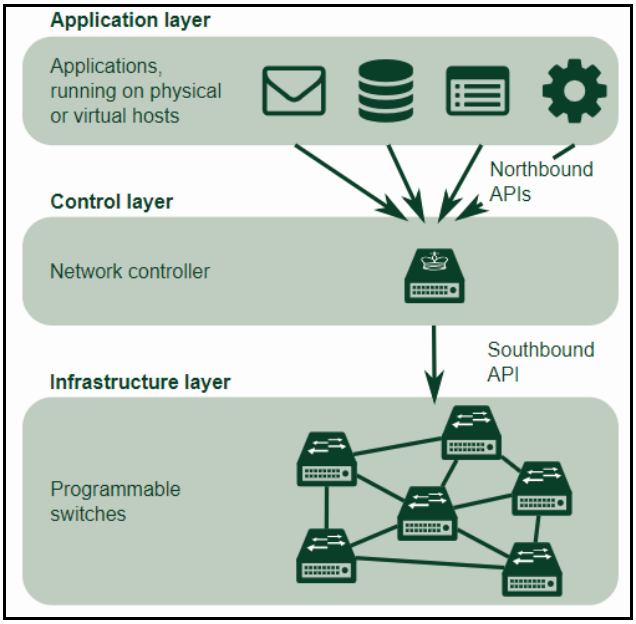


Figure 2. Architecture of SDN approach.

2.1.2. SDN controller

SDN Controller is where the administration of the network occurs. It is the "brain" of the network that corresponds to the setting up of the packet processing rules, and then the establishment of the entire network switching policy. In other words, SDN controller acts as a strategic point, which manages flow control towards the switches/routers below (via southbound APIs) and the applications and business logic above (via northbound APIs) to deploy intelligent networks. By taking the control plane off the network hardware and running it as software instead, the SDN controller facilitates automated network management and makes it easier to integrate and administer business applications.

An SDN controller platform typically contains a collection of pluggable modules that can perform different network tasks. Some of the basic tasks include inventorying what devices are within the network and the capabilities of each, gathering network statistics, etc. Extensions can be inserted that enhance the functionality and support more advanced capabilities, such as running algorithms to perform analytics and orchestrating new rules throughout the network. This means the actual packet processing. When packets require a particular and more complex processing, they can be handled by the SDN controller, where the decision regarding them will take place.

2.1.3. SDN benefits

The benefits of separating the control and the data planes are multiple and it is expected to have a great impact in the field of data networking. Some of them are denoted in the following:

- **Centrally Programmable Network**

  It allows network managers to configure, manage, secure, and optimize the network resources via SDN applications. Moreover, the network intelligence is

(logically) centralized in the SDN controller that maintains a global view of the network.

- **Precise Forwarding Control**

  In this case, SDN controller can apply its forwarding policies at a very granular level through an API.

- **Specialization**

  The separation of hardware and software can and it allows end users to select a combination of hardware and software that best suits their needs.

- **Open standards-based and vendor-neutral**

  This is due to the capabilities that are provided by SDN controller instead of vendor-specific devices and protocols, so it simplifies the network design and operation and at the same time it takes advantage of the SDN-based orchestration and management tools to quickly deploy, configure, and update devices across the entire network.

- **The higher rate of innovation**

  SDN will accelerate innovation and in this way, the abstraction of the network will allow building services on top of it without suffering from vendor lock-in while using common and standard interfaces.

- **Increased network reliability and security**

  Because SDN controller provides complete visibility and control over the network, it can enforce access control, traffic engineering, quality of service, security, and other policies.

- **Cost Reduction**

  By using SDN, the expenses will be driven into the networking software. As a consequence, hardware vendors will focus on reducing the cost of physical devices. Moreover, due to the software implementations, devices can be modified/upgraded through software rather than substituting them with new ones. This reduces CAPEX and also provides a simplified management which will result in a reduction of OPEX.

2.2. Network Function Virtualization (NFV)

Creating new services in current networks constitutes an important challenge, since these systems are vertically integrated, proprietary, and deployed mainly in hardware (along with the corresponding software). As such, they are difficult to be upgraded and managed and they lack of flexibility. This in turn means that modifications in both hardware and software have to be performed in order to be able to define new services leading to an increase in CAPEX and OPEX.

Networks` evolution provides the opportunity to avoid such issues by offering newly developed technologies that allow us to have more flexible and less manufacturer-dependent networks using open standards and at the same time with reduced CAPEX and OPEX.

Network Function Virtualization (NFV) [4] as a solution can cope with the abovementioned problems. NFV is a new IT virtualization technology that takes advantage of the advances in dynamic cloud architecture and SDN. NFV intends to shift network functions from specific hardware to software patterns that run on a general purpose virtualized networks. Figure 3, depicts the network virtualization concept.

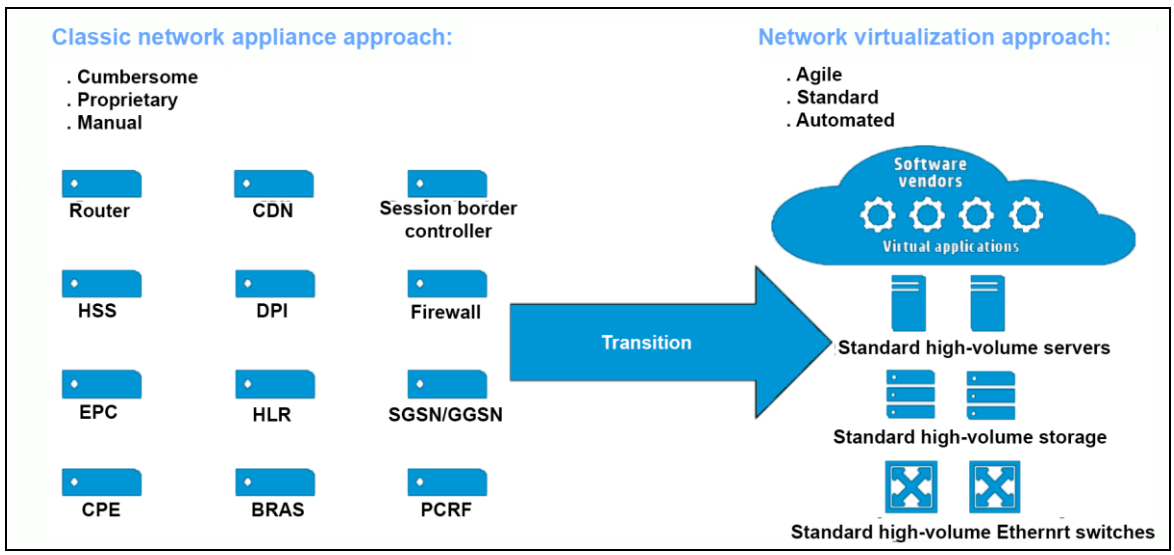


Figure 3. Network Virtualization concept.

By moving intelligence and workloads into software, end users are able to scale out dynamically and to meet changing traffic and service usage levels. In addition to this, NFV allows possessing multiple management points in the networks by using programmed software placed on different platforms. Therefore, it turns out that standardization and compatibility of these management points for multi-vendor solutions are applicable.

Based on what has been discussed so far, the principal features that characterize virtualized networks are as follows:

- **Dynamic service provisioning**

  Network operators can scale out dynamically the NFV performance on demand by having a precise granularity control according to the actual status of the network.

- **Decoupling hardware and software**

  By hardware and software separation the opportunity of independent evolution for each section arises.

- **Flexible deployment of network functions**

  NFV is able to implement automatically the network function software on a set of hardware resources that can execute several functions at different times in various datacentres.

2.2.1. NFV enablers

Recent technology developments make the goals of Network Function Virtualization achievable. The enablers of NFV as follows:

- **Cloud Computing**

  Network Function Virtualization will leverage modern technologies, such as those developed for cloud computing. The virtualization mechanism is at the core of cloud technologies: hardware virtualization using virtual Ethernet switches for connecting traffic between Virtual Machines (VM) and physical interfaces. For communication-oriented functions, high-performance packet processing is available through high-speed multi-core CPUs with high I/O bandwidth. In addition, smart Ethernet NICs can be used for load sharing, TCP Offloading, and routing packets directly to VM memory.

  Cloud infrastructures improve resource availability and usage by means of orchestration and management mechanisms. These mechanisms are responsible for the management of resources by assigning virtual elements to the correct CPU core, memory and interfaces.

  Finally, the availability of open APIs for management provides an additional degree of integration of Network Function Virtualization and cloud infrastructure.

- **Industry Standard High Volume Servers**

  The use of industry standard high volume servers is a key element in the economic case for Network Function Virtualization. An industry standard high volume server is a server built using standardized IT components (for example x86 architecture). A common feature of industry standard high volume servers is that there is competitive supply of the subcomponents that are interchangeable inside the server. There is a trend indicating that network appliances that depend on the development of bespoke Application Specific Integrated Circuits (ASICs) will become increasingly uncompetitive against general purpose processors as the cost of developing ASICs increases exponentially, while the feature size is decreasing [5].

2.2.2. NFV architecture

The NFV architecture is composed of the following sections, as shown in Figure 4:

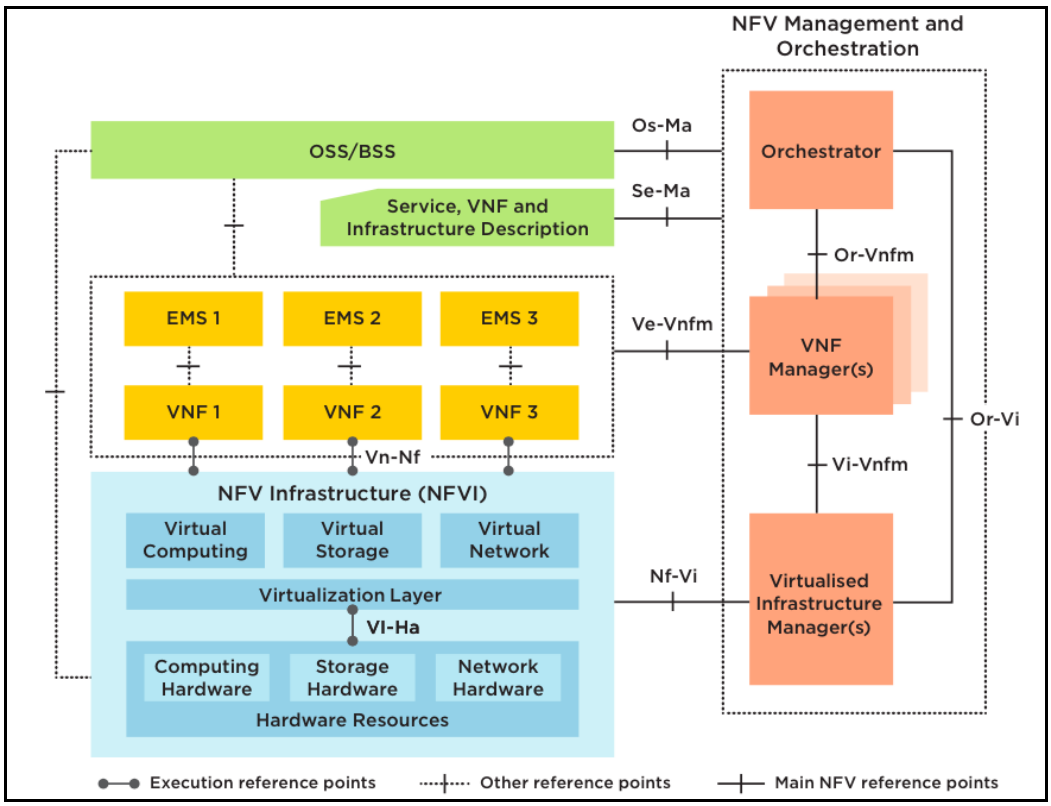


Figure 4. NFV architecture.

- **VNF (Virtualized Network Functions)**

  The implantation of NFV is done through the virtualization of network function elements called Virtualized Network Functions (VNF). Let us notice that, even when one sub-function of a network element is virtualized, it is also called VNF. A VNF creates a network function and when it combines different VNFs, a virtualized network segment is deployed. There are three types of devices that can be virtualized:

  - network function devices (routers, switches, Access Points, etc.).
  - network-linked storage (file servers and databases).
  - IT devices related to the network (firewalls, network device management systems, etc.).

- **EMS (Element Management System)**

  This is the element management (EM) system for VNF. The EM is responsible for the functional management of VNF, such as Fault, Configuration, Accounting, Performance and Security Management. The EM may manage the VNFs through proprietary interfaces.

- **VNF Manager (VNFM)**

  A VNF Manager, as the name suggests, manages a VNF or multiple VNFs. It performs the life cycle management of VNF instances. Life cycle management means setting up, maintaining and taking down VNFs. Let us point out that the EM is responsible for the management of functional components, while the VNFM manages the virtual components.

- **NFVI (Network Function Virtualization Infrastructure)**

  The environment in which VNFs are running is called Network Function Virtualization Infrastructure (NFVI). NFVI includes physical and virtual resources, as well as the virtualization layer.

- **VIM (Virtualized Infrastructure Manager)**

  VIM is the management entity for NFVI. It is responsible for controlling and managing the NFVI compute, network and storage resources within an operator's infrastructure domain. Moreover, it is responsible for the collection of performance measurements and events.

- **NFV Orchestrator**

  It generates, maintains and removes network services of VNFs. If there are multiple VNFs, the orchestrator will enable the creation of an end to end service over multiple VNFs. In addition, the NFV Orchestrator is also responsible for global resource management of NFVI resources.

- **OSS/BSS (Operation Support System/Business Support System)**

  OSS/BSS in NFV architecture refers to OSS/BSS of an operator. OSS deals with network management, fault management, configuration management and service management, while BSS deals with customer management, product management and order management. The current OSS/BSS of an operator may be integrated with the NFV Management and Orchestration using standard interfaces.

2.2.3. NFV benefits

The most significant benefits of using NFV are:

- **Lower CAPEX**

  The most obvious benefit of NFV is that it eliminates the need to purchase costly specialized appliances for networking functions. Instead, enterprises can meet up the specific needs by using commodity servers. These servers generally cost a small fraction of the price of specialized appliances and in most of the cases the network can be extended/modified by simply by adding more VMs to the already existing hardware of the infrastructure. Moreover, since enterprises need reliable and uninterrupted service from their networks, they build redundancy into their networks in case that a particular piece of hardware fails by moving services around to other hardware components as necessary.

- **Lower OPEX**

  By centralizing network management (especially if it is deployed alongside SDN), NFV can decrease the amount of time required to maintain networks. In addition, NFV deployments can be more energy efficient, resulting in lower utility costs.

- **Greater flexibility**

  Since there is no longer the need to buy expensive appliances for various functions, it becomes much easier for enterprises to deploy new functions. NFV also makes it easier and more affordable for companies to try out new network function technologies.

- **Improved scalability**

  Enterprises are able to grow their networks in an efficient way while using NFV. They no longer will need to buy costly hardware for various network functions. It means that adding more capacity to the network becomes much more affordable, and it requires much less time and work in the network administrators` part.

- **Enhanced security**

  In an NFV environment, adding new security technology to the network becomes much less expensive, making it easier to keep up with cybercrimes.

- **Improved user experience**

  Implementing NFV also results in a better experience for end users. In the case of hardware failure that otherwise could cause outages or slow performance, NFV makes it easy to move workloads to other hardware without requiring the purchase of additional expensive appliances. Moreover, the lower costs and improved scalability makes it easier to deal with the growing demand. So, mobile operators can have their service level agreements (SLAs) and as a result, end users don't have to worry about downtime because network performance still remains fast.

- **Faster provisioning**

  The ability to add new capabilities to the networks very quickly is one of the main factors that attracts service providers to shift to NFV. With NFV, service providers can provision new network features on the infrastructure that is already in place by using additional VMs to handle the additional workload.

- **Independence of manufactured equipment**

  The virtualized nature of NFV enables enterprises to have interoperability instead of sticking to some exclusive standards, due to the fact that there are different open standards that make it easier to have compatibility between several elements of the network.

2.2.4. Relationship between NFV and SDN approaches

As depicted in Figure 5, NFV is highly complementary to SDN in order to improve the functionality and performance of the whole system. However, it is worth of noting that they do not depend to each other. NFV has the possibility to be implemented without

SDN being required, as well as SDN without NFV. However, these two concepts and solutions when combined result in better network performance and higher flexibility since orchestration of a set of services as a significant peculiarity is a principle for both NFV and SDN.

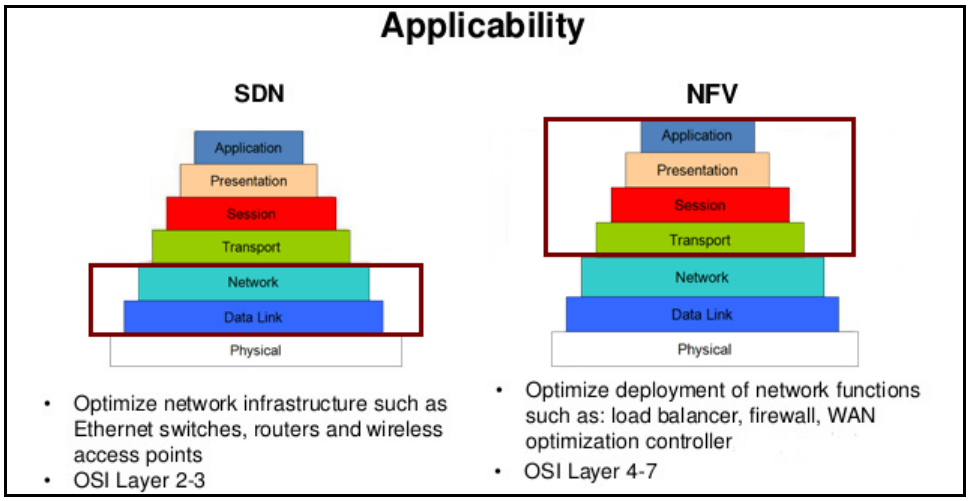


Figure 5. SDN and NFV applicability in different OSI layers.

Even though NFV goals can be also achieved when using non-SDN mechanisms, approaches that rely on the separation of the control and data planes, as proposed by SDN, can simplify compatibility with existing deployments, enhance performance, and facilitate operation and maintenance procedures. NFV is also able to support SDN by providing the infrastructure upon which the SDN can be run. Nowadays, in multi-provider environments that employ the scheme of management and monitoring tools of SDN, it turns out to be a necessity, and consequently it implies that management solutions between SDN and NFV are ideal for such situations. Furthermore, NFV aligns closely with the SDN objectives to use commodity servers and switches.

In fact, as illustrated in Figure 6, SDN and NFV both have the same objectives: innovation, creation, competitiveness and at the same time, reducing CAPEX, OPEX, space and power consumption.

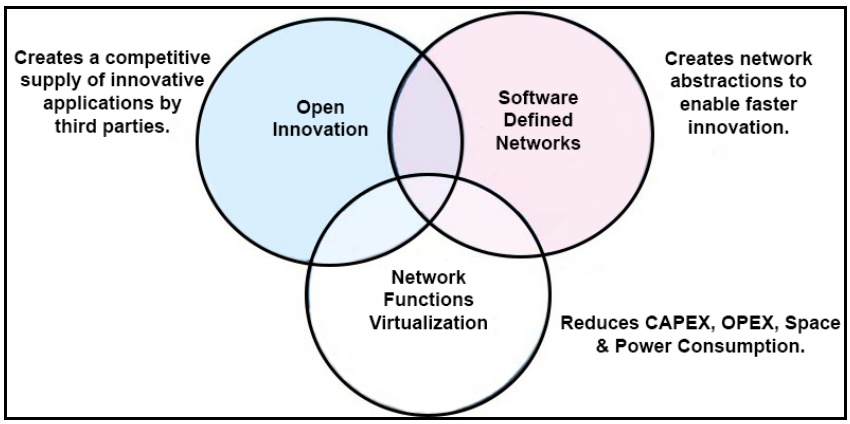


Figure 6. Relationship between SDN and NFV.

Of course it has to be mentioned that SDN is characterized by faster innovation, thanks to its network abstractions in the control and data planes.

2.3. Long Term Evolution (LTE)

As discussed in the Introduction chapter, in recent years the demand for exchanging data traffic with high quality through mobile networks has grown dramatically. It is due to the entry of smartphones into the market which in turn, increases the usage of mobile applications and consequently the internet. Since most of these applications have to work with the internet in order to offer better functionalities. According to these circumstances, communication standards have to be adapted in order to satisfy the technological and users` needs.

To address this high demand, LTE has been presented in the literature [10]. The LTE network is compatible with the last deployed technologies, GSM/UMTS, so the cost of deployment is relatively low and it can be implemented rather quickly. Let us notice, that GSM and/or UMTS are limited and most mobile operators are shifting towards the LTE standard.

2.3.1. LTE architecture

As illustrated in Figure 7, the LTE architecture is composed of the following components:

The Radio Access Network (RAN) and the Core Network (CN), which in LTE are called Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC), respectively. The E-UTRAN itself consists of one or a set of base stations, which are called evolved Node B (eNB). The EPC is also created by 4 entities: the Home Subscriber Server (HSS), the Mobility Management Entity (MME), the Packet Data Network Gateway (P-GW) and the Serving Gateway (S-GW). Moreover, the composition of Mobile Terminal (MT) and Terminal Equipment (TE) is called User Equipment (UE), which can be either SIM Card and cell phone or SIM Card and LTE dongle.

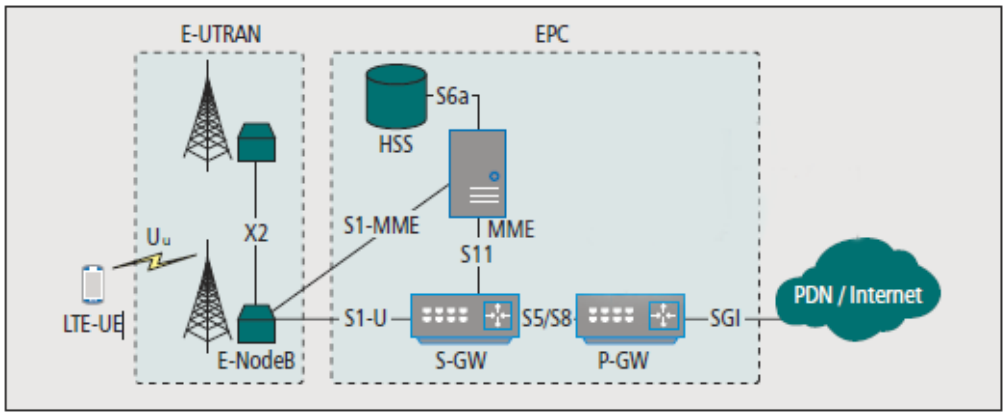


Figure 7. LTE architecture.

RAN (E-UTRAN)

Unlike the previous standards, RAN in LTE networks is characterized by its plain architecture without hierarchy and it consists of:

- **eNBs** which are in charge of all the access network functionalities such as [10]:
    - Radio Resource Management functions: Radio Bearer Control, Radio Admission Control, Connection Mobility Control, Dynamic allocation of resources to UEs in both uplink and downlink.
    - Measurement and measurement reporting configuration for mobility and scheduling
    - AS security.
    - IP header compression and encryption of user data stream.
    - Selection of a MME entity during the UE attachment procedure when no routing to a MME can be determined from the information provided by the UE.
    - Routing of User Plane data towards Serving Gateway.
    - Scheduling and transmission of paging messages (originated from the MME).
    - Scheduling and transmission of broadcast information (originated from the MME).
- **Uu** interface that enables the transferring of information by assigning radio channels to the connection between the UEs and the eNBs. The eNB on the other hand, implements all the functions and protocols needed in order to exchange information through the radio channels and it also controls the operation of the E-UTRAN interface.
- **S1** interfaces that allow the connection establishment between the UEs and the EPC. It is divided into two different interfaces: S1-MME and S1-U. By separating between S1-MME (S1-C) and S1-U, the separation between the control and data planes, respectively is achieved. As a result of this, their protocols are separated into two different stacks. The S1-MME is in charge of supporting the functions and procedures to manage the performance of this interface, while the S1-U is in charge of controlling the exchange of data in this interface. Therefore, the eNB and EPC communication will be divided into two parts [9]:
    - eNB with MME which is the main element in charge of the control plane in EPC. (see Figure 8)

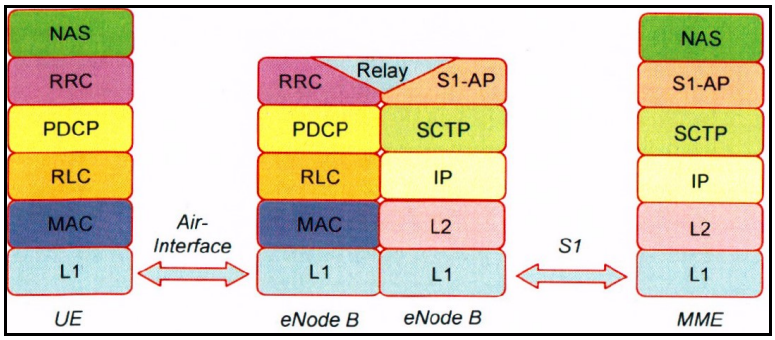


Figure 8. Control plane protocol stack.

- eNB with S-GW and consequently P-GW which are in charge of the data plane in EPC. (see Figure 9)

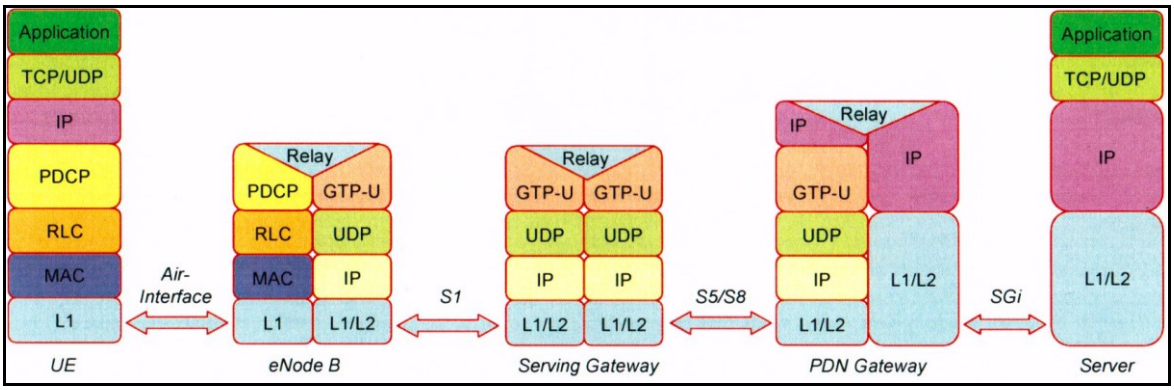


Figure 9. User plane protocol stack.

- **X2** is an optional interface which may be established between eNBs in order to exchange control and user information when needed. However, a full mesh is not mandated in an E-UTRAN network. Two types of information may typically needed to be exchanged over X2 to drive the establishment of an X2 interface between two eNBs [12]:
  - load- or interference-related information.
  - handover-related information.

CN (EPC)

EPC in LTE is in charge of providing IP connectivity services, regarding to its optimized architecture and includes the following entities [10]:

- **HSS**

  The information of UEs is stored in the HSS entity, which in fact is a database. Examples include the user subscription and network performance information. The HSS can be accessed by the MME via the S6a interface in order to control connectivity of service accessibility.

- **MME**

  It is the main element of the control plane in the EPC for managing the access of the UEs through the E-UTRAN. Every UE that is registered in the LTE network and is accessible through the E-UTRAN has an assigned MME entity. MME duties are:
  - Authentication and authorization of the users to access the network.
  - Location of the UEs in idle mode, i.e. those that are not active.
  - Radio Resource Control (RRC) connection in E-UTRAN.
  - Selection of S-GW and P-GW entities.
  - Support of functions for handover between eNBs and intra system handover (between E-UTRAN and other 3GPP networks).
  - Termination of the NAS (Non Access Stratum) signalling protocols.

- **S-GW**

  Provides the data plane communication between E-UTRAN and EPC. Every UE registered in LTE has an assigned S-GW entity in the EPC depending on the terminal geographical location and loads balancing criteria. It can be accessed, by the MME via the S11 interface, and by the P-GW via the S5/S8, interfaces. S-GW tasks are:

  - It is the local Mobility Anchor point for inter-eNB and inter-3GPP handovers.
  - It is in charge of users traffic routing and temporal storage of users IP packets from terminals that are in idle mode and initiates the network triggered service request procedure.
  - Performs user traffic routing to/from one or more P-GWs.
  - UL and DL charging per UE, PDN, and QCI.
  - Accounting and QCI granularity for inter-operator charging.
  - Control of use policies, control of accounting and lawful interception.

- **P-GW**

  It is the gateway that provides data plane connectivity with the external Packet Data Networks (PDN) via the SGi interface. After registering to LTE, a UE is assigned at least one P-GW. P-GW is in charge of:

  - Assignment of IP addresses to the UEs.
  - Anchor point for inter-system mobility management (between LTE and non-3GPP networks).
  - UL and DL service level charging, gating and rate enforcement.
  - Applies controlling rules for the service quality parameters from the data sessions established by the LTE network.

2.3.2. LTE characteristics and benefits

Thanks to the network architecture enhancement that is obtained by the next releases (i.e. LTE-Advanced and LTE-Advanced Pro), LTE turns out to be a suitable standard solution in order to handle data traffic in mobile communication networks. In the following, some of the main characteristics of LTE networks and their benefits are pointed out [13]:

- High total user capacity by applying robust OFDMA (Orthogonal Frequency Division Multiple Access) technique in the downlink for dealing with multiple interferences.
- SC-FDMA (Single Carrier-FDMA) technique in the uplink, which leads to low power during transmission.
- By applying Multiple-Input and Multiple-Output (MIMO) antenna technique along with OFDM, higher data rates and much better SNR at the receiver can be achieved, especially in dense urban areas.
- The upper layers of LTE are based on TCP/IP, which results in an all-IP network, so it is able to support packet and data services.
- Separation of the data plane and the control plane through open interfaces.
- Very low latency for both control and data planes providing a better network performance.

- The capability of operating in different frequency bands resulting in increment of both bandwidth and data transmission rates.
- Improvement and flexibility of the use of the spectrum (FDD and TDD) making a more efficient management of it including unicast and broadcast services.

2.3.3. Implementing of SDN and NFV over a LTE network

Despite the positive impacts of the LTE network in data traffic transmission, an immense increase in the amounts of data traffic is expected for the next few years that most probably will lead to congestion and overflow of the network. Accordingly, mobile network operators and researchers are focusing on the development of solutions that will be able to handle the predicted data traffic levels.

By exploiting the advantages of SDN and NFV (such as virtualization, as well as the flexibility in the network management), a cellular network architecture called Cellular SDN (CSDN) has been proposed, depicted in Figure 10. This architecture enables network operators to simplify network management and control. It also enables the creation of new services in a flexible, open, and programmable manner.

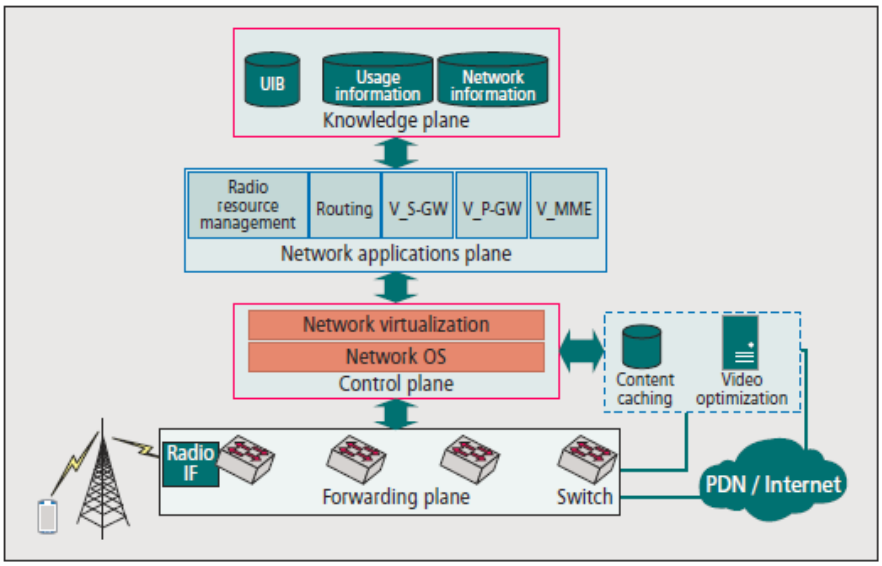


Figure 10. Cellular SDN architecture.

CSDN leverages the benefits of SDN and NFV as follows:

- The SDN platform, by applying orchestration of intelligent services, enables Mobile Service Provider (MSP) to implement subscriber policy, profile-aware service provisioning at the level of individual flows, etc.
- The NFV platform, which in fact is a cloud-based RAN (C-RAN) that allows dynamic resource management using virtualization techniques. It uses different virtual machines running multitudes of applications over the cloud infrastructure of an MSP. An application within this framework, for instance the MME can not only run on virtual machines, but can also adapt its capabilities in an elastic way

depending on the network load. This facilitates the network dimensioning and resource allocation. [9]

2.4. Real-time SDN-NFV platform

It is obvious that to effectively manage the communication network, a platform is needed in order to implement SDN-NFV solutions. It provides an abstraction layer for switching/networking and computational operations in order to share resources of different domains for successful delivery of end-to-end services according to the well-organized scheduling task. The real-time SDN-NFV platform makes it easy to monitor, control and initialize the operations that are necessary. In the following the three entities which construct this platform will be explained.

2.4.1. OpenAirInterface LTE (OAI LTE) platform

Software-Defined Radio (SDR) is a system that implements radio equipment (hardware) on software basis using low-cost general purpose computers and radio frontends. OpenAirInterface LTE (OAI LTE) platform is the first open-source software-based implementation of the LTE system that spans the full protocol stack of 3GPP standard both in E-UTRAN and EPC [14]. In the OAI LTE platform, the transceiver functionality is realized via a software radio front end connected to a host computer for processing. It can be used to build and customize an LTE base station and core network on a computer, connect commercial UEs to test different configurations, setup the network and monitor the network and mobile device in real-time. OAI provides a rich development environment with a range of build-in tools, such as highly realistic emulation modes, soft monitoring and debugging tools, protocol analyzer, performance profiler, and configurable logging system for all layers and channels [15].

2.4.1.1. OAI LTE platform-software section

As illustrated in Figure 11, the OAI LTE protocol stack includes three blocks: OAI soft UE, OAI soft eNB as E-UTRAN, and OAI soft MME+S+P-GW as EPC. The advantage of this partitioning targets to provide the possibility to perceive the control and data planes` separation by analysing the layers of the 3GPP standard.

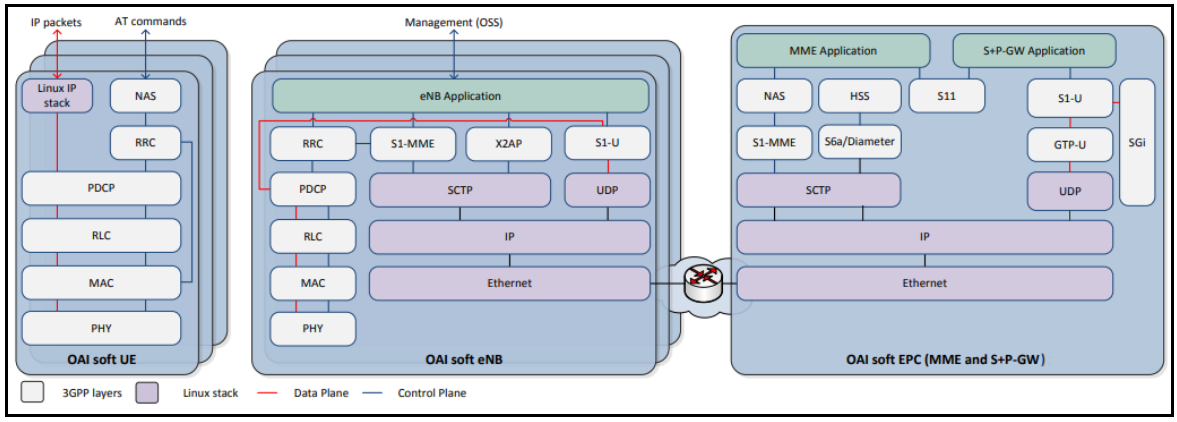


Figure 11. OpenAirInterface protocol stack.

OAI LTE platform includes a full software implementation of the LTE standard [14], [15].

At the **Physical layer**, it provides the following features:

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

For the **E-UTRAN protocol stack**, it provides:

- LTE release 8.6 compliant and a subset of release 10 features.
- Implements the MAC, RLC, PDCP and RRC layers.
- Priority-based MAC scheduler with dynamic MCS selection.
- Fully reconfigurable protocol stack.
- Integrity check and encryption using the AES algorithm.
- Support of RRC measurement with measurement gap.
- Standard S1AP and GTP-U interfaces to the Core Network.
- IPv4 and IPv6 support.

For the **EPC**, it provides:

- MME, SGW, PGW and HSS implementations.
- NAS integrity and encryption using the AES algorithm.
- UE procedures handling: attachment, authentication, service access, radio bearer establishment.
- Transparent access to the IP network (no external Serving Gateway nor PDN Gateway are necessary).
- Configurable Access Point Name (APN), IP range, DNS.
- IPv4 and IPv6 support.

Based on the above it is clear that most of the LTE network architecture entities, along with their corresponding interfaces and protocol stacks, have been implemented in

software in the OAI platform. In Figure 12, the OAI LTE architecture of the software section is shown:

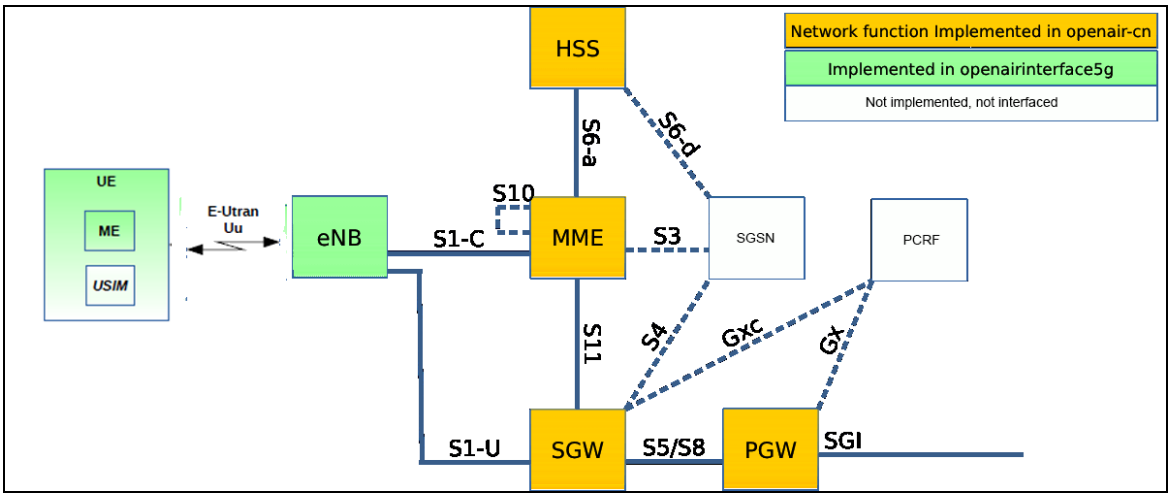


Figure 12. OAI LTE architecture-software section.

As Figure 12 indicates, OAI LTE-software section is split into two subsections:

- **openairinteface5g** (written in C/C++) plays the role of the E-UTRAN part of the LTE network. It gives the possibility to be customized in order to be used with other platforms, as for example with the 5G-EmPOWER platform (introduced in third part of this section).
- **openair-cn** (written in C/C++) virtualizes the four network entities (and their functions) of the EPC: HSS, MME, S-GW, and P-GW. In this case, S-GW and P-GW have been merged together in the SPGW virtualized entity so the S5/S8 interface has not been implemented. Same as previously, these three entities are able to be customized according to the user's requirements.

2.4.1.2. OAI LTE platform-hardware section

Apart from software, hardware components are necessary in order to create an interface (Uu) and then establish the connection between UE(s) and the eNB in E-UTRAN for having real-time communication. OAI LTE has been successfully tested with EXPRESSMIMO2 embedded system board in real-world experimentation, and also Universal Software Radio Peripheral (USRP) B2x0 created by Ettus Research/NI. The latter is widely used in the research community. For the implementation of the Uu interface in this project, the USRP B210 has been used (see Figure 13).

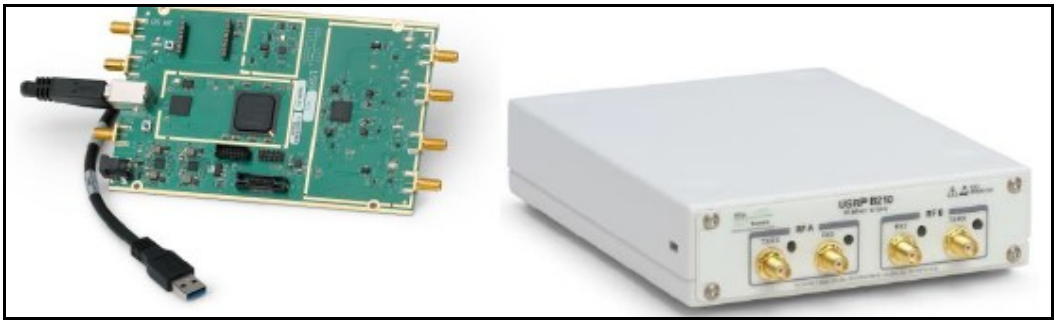


Figure 13. USRP B210, Ettus Research.

Some of the specifications of USRP B210 are as follows [17]:

- Radio frequency coverage from 70 MHz to 6 GHz.
- USB 3.0 SuperSpeed interface and Standard-B USB 3.0 connector.
- MIMO 2TX & 2RX capability.
- Half and Full Duplex.
- Open and reconfigurable Xilinx Spartan 6 XC6SLX150 FPGA.
- Up to 56 MHz of real-time bandwidth in 2TX & 2RX.

Moreover, 2 antennas for transmitting and 2 antennas for receiving (connected to the USRP device) are required for the transmission and reception of information between the UEs and the LTE network using MIMO 2x2. Typical Wi-Fi antennas can be used for this implementation. The USRP B210 and the antennas, will interact with a PC (Intel x86) running the software applications of E-UTRAN and EPC entities. Finally, a USB 3.0 cable is needed in order to connect the USRP B210 to the PC.

2.4.2. Software Radio Systems LTE (srsLTE) platform

As discussed in the previous section, OAI is the first and also most popular open source LTE SDR software available for testbeds. Software Radio System LTE (srsLTE) platform on the other hand, is another possibility in this area, since OAI LTE code structure in some cases is complex and difficult to be customized by a user external to the project [18]. It has to be pointed out that there are different options in which OAI platform can be used. Several configurations which can involve commercial components have been listed in the following:

- OAI UE ↔ OAI eNB + OAI EPC.
- OAI UE ↔ OAI eNB + Commercial EPC.
- OAI UE ↔ Commercial eNB + OAI EPC.
- OAI UE ↔ Commercial eNB + Commercial EPC.
- Commercial UE ↔ Commercial eNB + OAI EPC.
- Commercial UE ↔ OAI eNB + Commercial EPC.
- Commercial UE ↔ OAI eNB + OAI EPC.
- Commercial UE ↔ OAI EPC + srsLTE eNB.

So it is possible to customize the network structure by using various platforms. The last option (Commercial UE ↔ OAI EPC + srsLTE eNB), is the one that has been selected for this project. In the following, the srsLTE platform and in particular the srsLTE eNB is introduced.

2.4.2.1. srsLTE platform-software section

srsLTE is a high-performance SDR for creating appropriate applications and it also has high modular, with minimal inter-module or external dependencies, software code. Moreover, it is available under both commercial and open-source licenses. srsLTE software section is split into the following subsections:

- **lib** (written in C/C++) is a modular and portable library for srsLTE, as illustrated in Figure 14. lib aims to provide the tools to build LTE-based applications, such as a complete eNB, EPC, UE, an LTE sniffer or a network performance analyzer, since L1, L2 and L3 protocol stacks for UE and eNB have been implemented in this library.

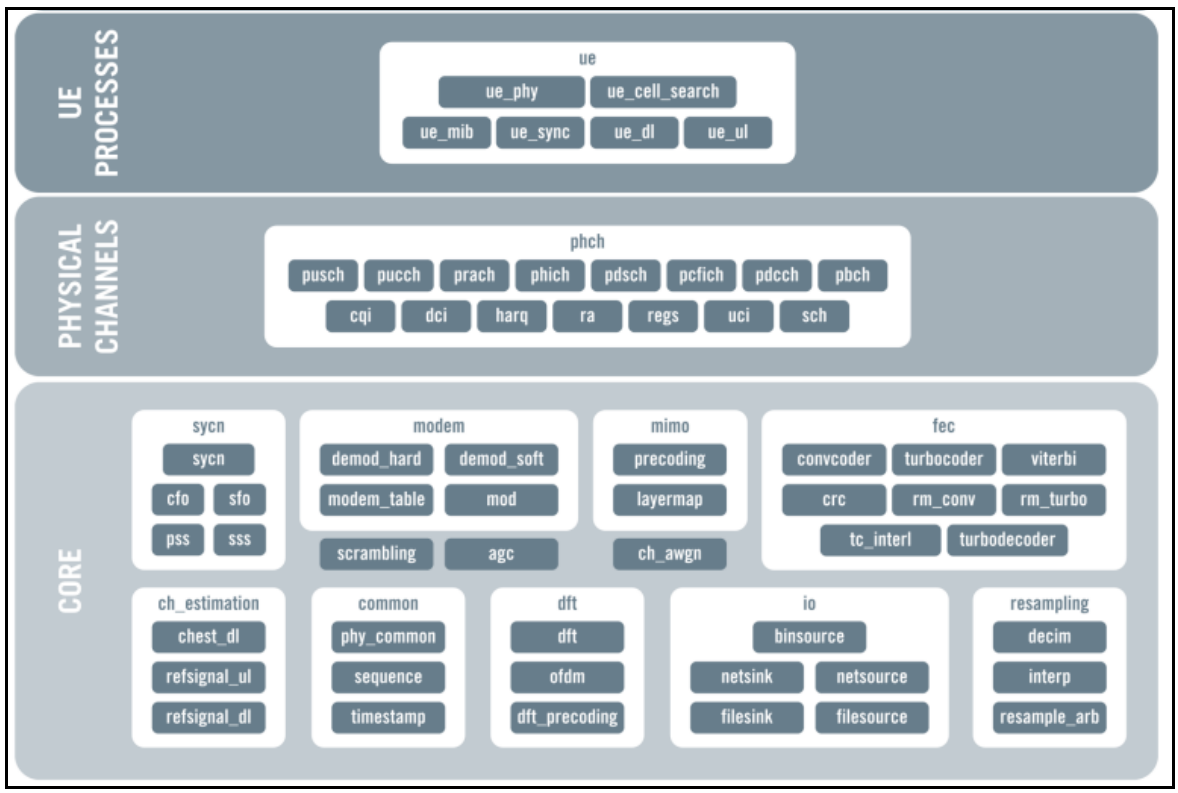


Figure 14. Module diagram for the srsLTE library (lib).

Some of the current features provided by this library are [18]:

- LTE Release 8 compliant, with FDD configuration.
- Supported bandwidths: 1.4, 3, 5, 10 and 20 MHz.
- Transmission mode 1 (single antenna) and 2 (transmit diversity).
- Evolved multimedia broadcast and multicast service (eMBMS).
- MAC, RLC, PDCP, RRC, NAS, S1AP and GW layers.

- **srsue** (written in C/C++) is the application created based on lib, which virtualizes the UE processes by implementing the physical channel procedures for uplink and downlink making use of the physical channel modules (from lib).
- **srsepc** (written in C/C++) is single binary, light-weight LTE EPC implementation which contains all four entities of LTE EPC (HSS, MME, S/P-GW) in virtualized forms.
- **srsenb** (written in C/C++) is an implemented application based on lib, which has the complete features of an LTE eNB.

As such, based on the library modularity, custom system design and development on a range of SDR platforms can be achieved, as in this project, by implementing the network with: Commercial UE ↔ OAI EPC + srsLTE eNB.

srsLTE eNB

srsLTE eNB (**srsenb** code) is a complete software radio LTE eNB built upon the srsLTE library and that can run on a PC or laptop. srsLTE eNB supports full uplink and downlink rates using a standard 20 MHz carrier and is available in source and binary commercial license upon request. Figure 15, depicts the srsLTE eNB protocol stack:

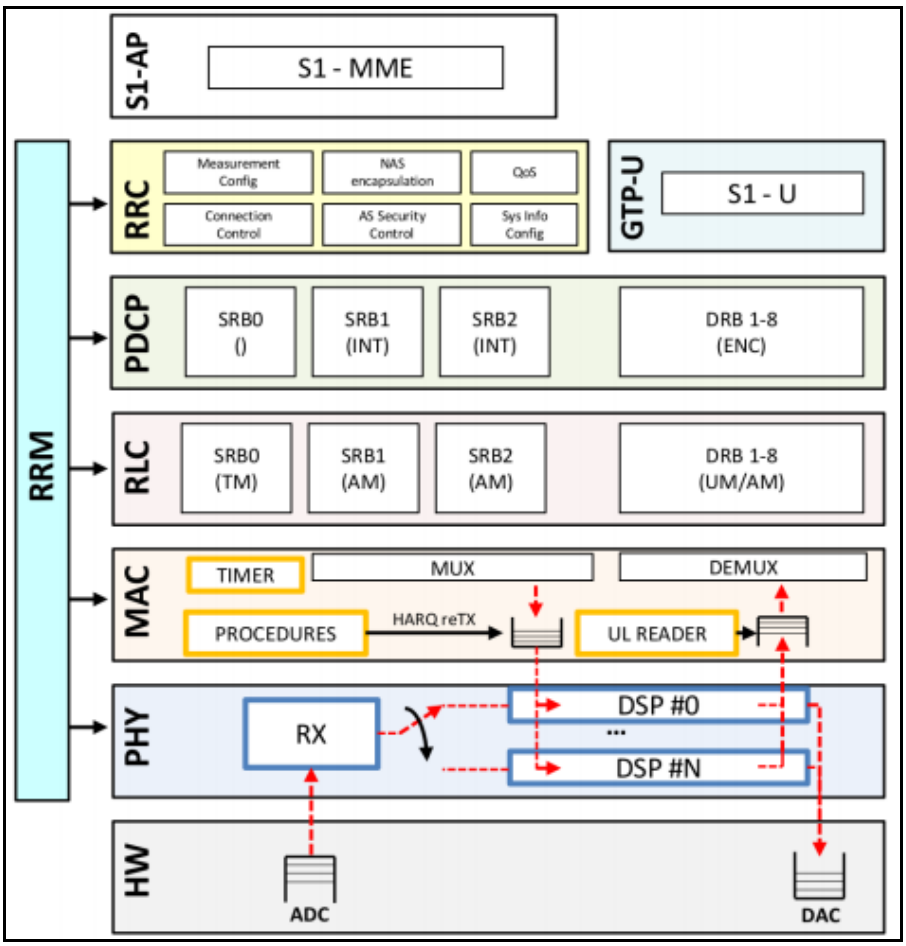


Figure 15. srsLTE eNB protocol stack.

Apart from the previously mentioned features of the srsLTE library, srsLTE eNB also has the following characteristics [19]:

- Tested up to 75 Mbps DL in SISO configuration with commercial UEs (LG Nexus: 4 and 5, Huawei dongles: E3276 and E398).
- Round Robin MAC scheduler.
- PUCCH Format1 and Format1A receiver.
- All DL channels/signals are supported for eNB side: PSS, SSS, PBCH, PCFICH, PHICH, PDCCH, PDSCH.
- Standard S1AP and GTP-U interfaces to the Core Network.
- Periodic and Aperiodic CQI feedback support.
- 150 Mbps DL in 20 MHz MIMO and 75 Mbps DL in SISO configuration with commercial UEs (LG Nexus: 4 and 5, Huawei dongles: E3276 and E398).

And in the User Interfaces, srsLTE eNB provides:

- Detailed log system with per-layer log levels and hex dumps.
- MAC layer Wireshark packet captures.
- Command-line trace metrics.
- Configuration file for SIB1, SIB2, DRB and Radio Resource configuration.

2.4.2.2. srsLTE platform-hardware section

Similarly, as in the OAI LTE platform-hardware case, the need for proper hardware in order to create the Uu interface in the E-UTRAN part is undeniable. srsLTE library deals with buffers of samples in system memory, thus it is able to work with any RF front-end. It currently provides interfaces to the Universal Hardware Driver (UHD), giving support to the Ettus USRP family of devices. So, USRP B210, the number of antennas required and also a PC (Intel x86) running the software application of the E-UTRAN (in this case srsLTE eNB), constitute the necessary hardware equipment for the implementation.

2.4.3. 5G-EmPOWER platform

5G-EmPOWER Controller (also known as EmPOWER) is an open multi-access network operating system. It has been built upon a single platform that consists of general purpose hardware (x86) and operating system (Linux). 5G-EmPOWER, by introducing the programmable data plane concept, delivers three types of virtualized network resources: forwarding nodes (OpenFlow switches), packet processing nodes (Micro Servers), and radio processing nodes (Wi-Fi Access Points or LTE eNBs). With its flexible architecture and the high-level programming APIs, 5G-EmPOWER enables fast prototyping of applications and services. Some of the 5G-EmPOWER platform specifications are summarized in the following [20]:

- Virtualized radio access network that supports both LTE and Wi-Fi technologies.
- Centralized mobility management for LTE and Wi-Fi.
- Full visibility of the network status and dynamic deployment, management, and orchestration of the network services.
- Implementation of user resource allocation schemes within a network slice
- Supports multi-tenancy which in turn allows the implementation of different slices of applications and services from different operators or tenants over the same infrastructure.

- Customized packet processing in a portion of the traffic.
- Web-based control framework.

In Figure 16, the architecture of the 5G-EmPOWER network is presented.

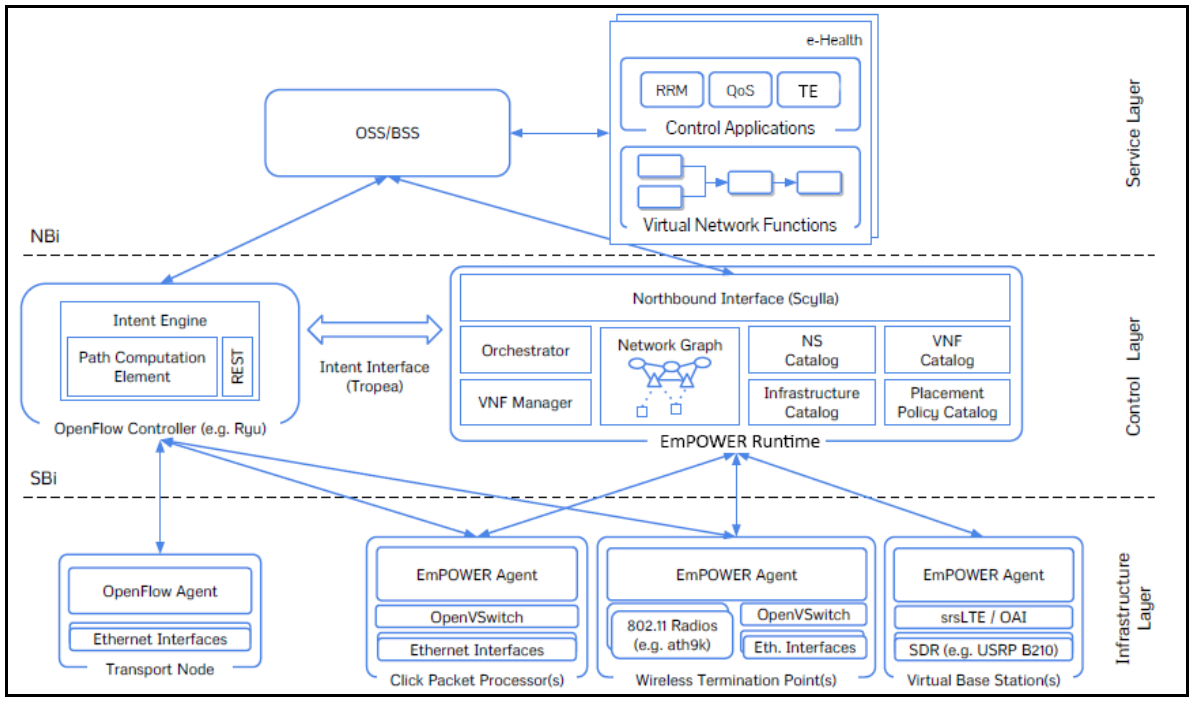


Figure 16. 5G-EmPOWER architecture.

As it is can be seen from the Figure 16, 5G-EmPOWER consists of three layers: Service, Control, and Infrastructure.

- At the **Service layer**, the network services are located, which essentially include a set of VNFs and their associated control logic. In fact, they are the network applications that are running in their own slice of resources and communicate with the Controller layer via the Northbound interface (REST API or a native Python API).
- At the **Control layer**, there are the EmPOWER Runtime (Controller) and a wired backhaul controller (e.g. OpenFlow). They interact with each other using the intent-based interface. The EmPOWER Runtime is responsible for translating the high-level control policies specified at the Service layer, such as Radio Resource Management and Traffic Engineering, into commands for the data plane elements (Infrastructure network devices). It has to be pointed out that the EMPOWER Runtime utilizes an Agent interface (EmPOWER Agent) for its communication with the infrastructure network devices though the southbound interface. A controller-agent model is appropriate for the communication between a controllable and a controlling entity and applies recursively to the SDN-NFV architecture. The controllable entity aims to use the agent, in order to represent its resources and capabilities to the controlling entity with a wrapper that is used to translate the exchanged information. Even though the agent's location is inside the controlling entity domain (in an SDN-NFV controller platform like 5G-EMPOWER), the agent

notionally resides in the controllable entity domain (srsLTE eNB in this project). In particular, EmPOWER Runtime treats radio access as a VNF and so, it is in charge of deploying the Light Virtual Access Point (LVAP) and Light Virtual Network Function (LVNF) on the network devices at the Infrastructure layer. In addition, multi-tenancy aspect on the top of the same physical infrastructure is one of the main capabilities which it can afford. A tenant is a virtual network with its own Infrastructure network devices (Click Packet Processors (CPPs), Wireless Termination Points (WTP), or Virtual Base Stations (VBSes)). Finally, the EmPOWER Runtime ensures that a network application will receive only a perspective of the network corresponding to its own slice. Some of the main features of the EmPOWER Runtime are listed as follows:

- **Soft State**

  List of network slices and client's authentication method are the only persistent information stored in the controller, while others are deleted when the network is disconnected from the controller. Moreover, the network itself can still function at its last known state even if the controller becomes unavailable [20].

- **Modular Architecture**

  Every task supported by the controller, except the logging subsystem, is deployed as a Python-based plug-in module, which can be loaded at runtime, such as implementing the data-path control protocol.

- **Slicing**

  Multiple logical virtual networks or slices can be created on top of the controller. Each slice will be distinguished by its own network name and set of WTPs for Wi-Fi and VBSes for LTE networks. Applications run on one or more slices and users can associate with a slice based on its network identifier.

In the case of the backhaul controller (OpenFlow Controller), it uses the OpenFlow Agent though the southbound interface in order to communicate with the backhaul Transport Nodes [21].

- At the **Infrastructure layer**, data plane elements are placed:
  - Wireless Termination Points (WTPs) or Wi-Fi Access Points, the physical points of attachment in Wi-Fi radio access network.
  - Click Packet Processors (CPPs), the forwarding nodes with packet processing capabilities.
  - Virtual Base Stations (VBS) or LTE eNBs, the physical points of attachment in LTE radio access network.

2.4.3.1. 5G-EmPOWER-software section

It is split into the following subsections:

- **empower-runtime** (written in Python) is the actual part of the 5G-EmPOWER controller that manages the network.

- **empower-enb-proto** (written in C/C++) is a library containing the protocol definitions and parsers used to format or decode the exchanged messages between the with the EmPOWER Agent and the EmPOWER controller. This component is usually included in the EmPOWER Agent. [22]
- **empower-enb-agent** (written in C/C++) implements the EmPOWER Agent. The Agent contains the different types of actions that can occur in the system (based on events) and the routines that handle the exchanged information with the controller. Its main duty is to schedule the proper action to be performed as response to an event coming from the controller. In order to perform the various actions, the Agent needs to communicate with the srsLTE eNB, therefore it is located on the same physical machine (PC). Let us notice that for the communication with the srsLTE eNB a translator (wrapper) is needed in order to translate the instructions to the LTE stack.
- **empower-srsLTE** (written in C/C++) as discussed before, implements a customized version of the srsLTE eNB entity that integrates the srsLTE with the 5G-EmPOWER. This version includes the wrapper presented above that is responsible for the communication of the EmPOWER Agent with the srsLTE eNB. As such, the empower-srsLTE contains: the (srsenb+lib+srsue+srsepc) +agent, however let us notice that the srsue and srsepc modules are not used in this project.

Figure 17, depicts a complete block diagram of the three entities (OAI LTE, srsLTE eNB and 5G-EmPOWER) that consist the real-time SDN-NFV platform of the project, from both software and hardware perspectives.

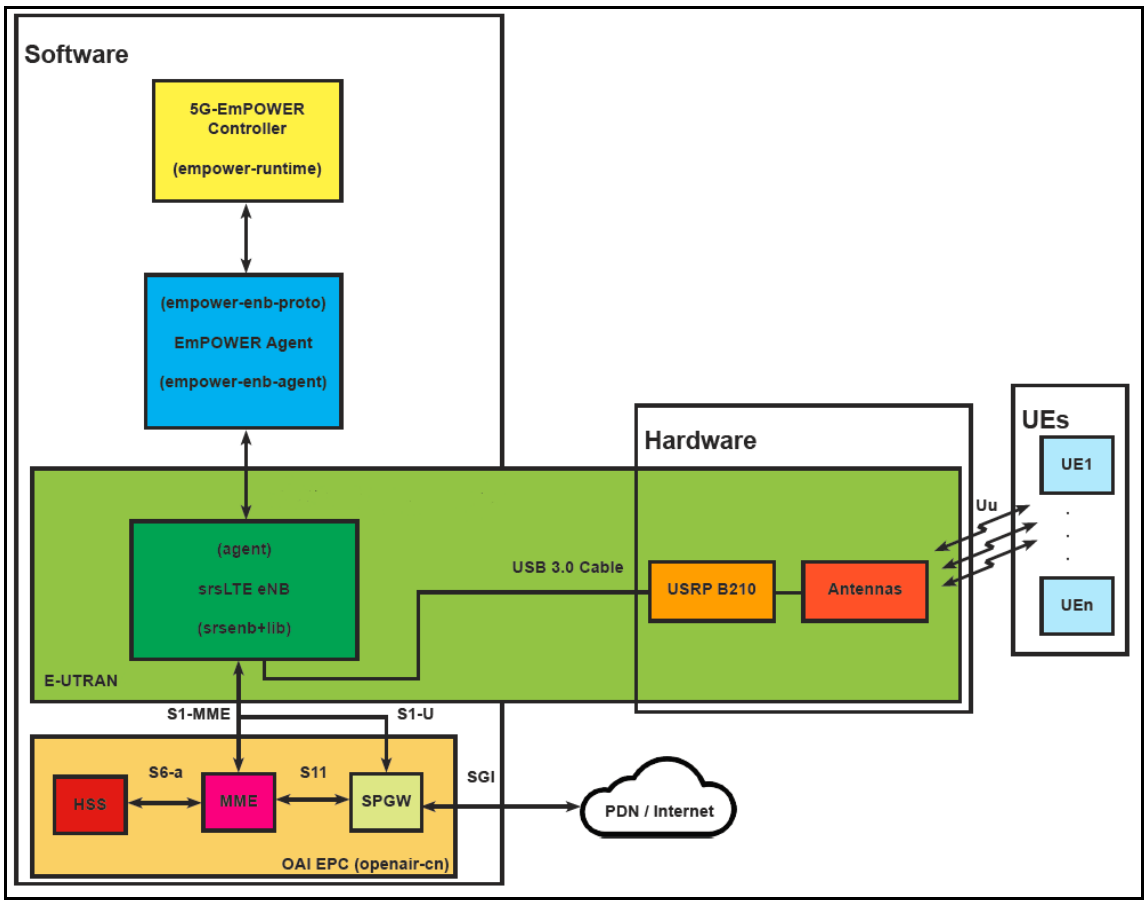


Figure 17. Real-time SDN-NFV platform of the project (software and hardware perspectives).

3. Methodology

To understand the objective of the project, which is the implementation of a RAN slicing technique over the LTE network in a real-time SDN-NFV platform, it is necessary to perform several steps in order to achieve the desirable results. In this chapter, these steps are briefly explained; however, the alerted reader may refer to the tutorials created during this project for a step by step description of the procedure that has been followed.

3.1. Study of the real-time SDN-NFV platform

As discussed previously, the real-time SDN-NFV platform in this project is composed of the OAI LTE EPC as core network, the srsLTE eNB as E-UTRAN and the 5G-EmPOWER as the controller. The main stages that have been followed in this phase of the project to get familiarized with these platforms are:

- Study of the main characteristics of the OAI LTE EPC, srsLTE eNB and 5G-EmPOWER controller, as well as of the different entities that are involved in each section.
- Familiarization with the installation, compilation and debugging processes of these entities and the integration with each other.
- In the case of the 5G-EmPOWER, the study has focused on the information messages that can be monitored in the PC terminal when executing it, on what functionalities the controller web interface has, and how this interface can be used in order to perform RAN Slicing.
- Handling and addressing common errors that might occur at each stage. This process included online investigation of possible solutions that have been proposed from different research groups working on similar projects.

3.2. OAI EPC platform implementation and database creation

It is feasible for OAI EPC (openair-cn code) to be installed and implemented either on a Physical Machine (PM OAI EPC) or on a Virtual Machine (VM OAI EPC). In this project, both implementations of OAI EPC have been tested and no differences have been observed in the two implementations. After installation and compilation, the different entities belonging to the OAI EPC (HSS, MME, and SPGW) have to be configured for both PM and VM by using several configuration files each entity has. This process is described in detail later on. Before proceeding to the configuration process the next steps must be taken under consideration:

- Firstly, some special parameters have to be defined. These parameters are extracted from the radio network regulations, which will differ according to each country, each mobile network operator inside the country and also various geographical areas each mobile network operator covers (General network parameters). Therefore, a new mobile network operator has to be created for this project based on a proper selection of these parameters.
- Secondly, customized SIM Cards have to be programmed according to their specifications, in order to enable them to connect to the mobile network operator.

These SIM Cards have been manufactured for performing connectivity tests in such networks. The same network parameters previously defined have to be programmed in the SIM Cards.

- Apart from the general network parameters` definition and the SIM Card programming, each mobile network operator has a database (located at the HSS entity) that stores the information of the SIM Cards that are allowed to be registered on the network. Let us notice that the database in the OAI HSS has been implemented with MSQL.

The parameters to be set in the OAI EPC configuration files are:

- General network parameters (such as MNC, MCC, etc.).
- Parameters used for the interaction with MySQL database.
- The interfaces and port numbers for the interaction between the different OAI EPC entities.
- IP addresses that have to be set on these interfaces.

3.3. srsLTE platform implementation and integration with OAI EPC platform

A similar procedure, but based on the instructions related to srsLTE, has to be followed in order to install the srsLTE on a (separate from the OAI) Physical Machine. Let us notice that in this project the empower-srsLTE eNB (provided by FuN-CreateNet) has been used that includes the EmPOWER Agent. Thus, the *empower-srsLTE eNB* version has been installed. Notice that in order to enable the Agent wrapper (the software needed for the translation of the instructions between eNB and Agent) the software has to be compiled with the *EmPOWER Agent* option enabled. Finally, it has to be mentioned that even though the whole srsLTE software will be installed, only the srsLTE eNB section will be used (executed) in this project.

The next steps will include the configuration of the srsLTE eNB through the provided configuration file (i.e. enb.conf). The following parameters have to be included in this file:

- The same general network parameters as in the OAI EPC case (e.g. MNC, MCC, etc.).
- The IP addresses to communicate with the OAI EPC (PM or VM) and also the IP address of the PC that runs the 5G-EmPOWER controller.
- The Port number of the 5G-EmPOWER controller in order to interact with it.

3.4. Mobile Equipment configuration

A Mobile Equipment (ME) needs to be configured in such a way so that it will be able to connect to the created mobile network using the SIM Cards programmed before. After that, some connectivity tests can be performed in order to verify that the network works properly. Moreover, the network performance can be tested for different QoS based on various services, such as web browsing and video streaming. These tests can be carried out with both cell phones and dongles.

3.5. Launching the network and performing connectivity tests

After the configuration process is accomplished successfully and ME configuration, the network is ready to be launched. Specifically, first the OAI EPC entities and then the srsLTE eNB are executed allowing us to verify that all the previous phases have been carried out correctly. After that, some connectivity tests can be performed in order to verify that the network works properly. Moreover, the network performance can be tested for different QoS based on various services, such as web browsing and video streaming. These tests can be carried out with both cell phones and dongles.

3.6. 5G-EMPOWER platform implementation and integration with srsLTE eNB

For integrating the whole EmPOWER system the procedure that has to be followed is based on the corresponding instructions of the 5G-EmPOWER Wiki. Let us notice that the controller can be installed either on a Virtual Machine, or on a Physical Machine. In addition, the EmPOWER Agent (responsible for the execution of different operations) and the EmPOWER protocol (the definition of the protocol used for the communication between the Agent and the Controller) must be installed too. Let us notice the code of the Agent is included in the empower-enb-agent code, while the protocol in the empower-enb-proto code. Finally, a configuration file for EmPOWER Agent that contains the IP address and the port number of the Controller has to be created.

In this way, the 5G-EmPOWER and the srsLTE eNB can be integrated. The last part in order to have a complete functional platform is to introduce the srsLTE eNB, (in the EmPOWER terminology it is called Virtual Base Station: VBS), to the 5G-EmPOWER Controller through the web interface it provides.

3.7. Implementation of a tenant slice and performing connectivity tests

The last step of the followed methodology is to implement a slice (or tenant) and connect various UEs to that slice (see Figure 18). This can be carried out using the 5G-EmPOWER web interface and by introducing the related parameters such as the PLMN Id (MNC+MCC). After this process is completed successfully, further connectivity tests with the UE(s) have to be performed in order to verify the correct operation of the network.

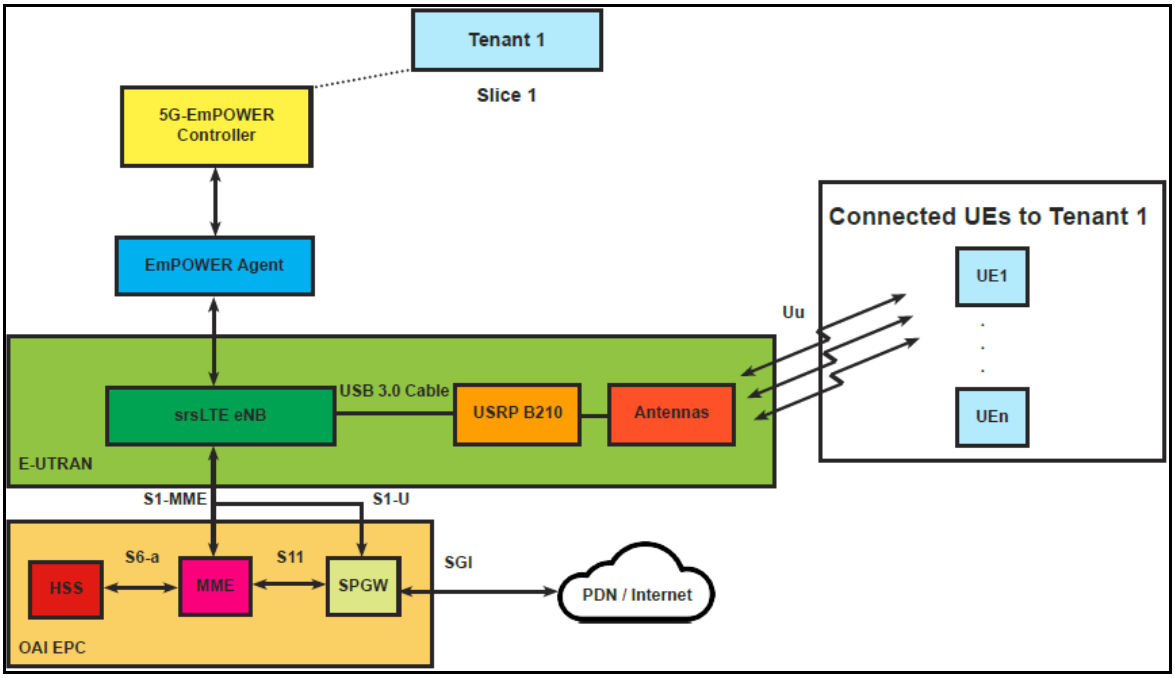


Figure 18. RAN Slicing in Real-time SDN-NFV platform in the project.

Let us refer that the current version of the empower-srsLTE does not support Multi-Operator Core Network (MOCN), which means that it cannot work with several different EPCs from different mobile network operators at the same time. It is very likely that such this important capability will be added in the future releases of this platform.

3.8. Network operation in debugging mode

There is a possibility for the network to operate in debugging mode. Operation in debugging mode provides the opportunity to analyze the network processes with detailed information. In this way, a more explicit idea about how different entities exchange their information can be obtained. One of the main advantages of working in debugging mode is to trace possible errors that may appear in the network in order to fix them. To do so, the debugging mode has to be enabled for the entities that offer this capability and the compilation process has to be executed again for making the system to accept the changes.

3.9. Code block diagram and code modifications

After finishing the previously presented phases and gaining proper knowledge of the whole network, the last part of the project includes the study of the code, the introduction of a new function and the validation of it using the debugging process. In that respect, the code has been studied in detail and a block diagram of the different code sections has been created. Finally, a new message has been implemented between the EmPOWER Agent and the Controller.

4. Development and achieved results

This chapter describes the implementation of the various procedures carried out during the development of this project in order to integrate the involved platforms and to achieve the main objective, which is to contribute to the implementation of RAN Slicing on a real-time SDN-NFV platform. In each section, the different results obtained from each process are presented and analyzed. Moreover, the system integration and the network operation in debugging mode are described. Finally, in the last section, the srsLTE eNB code block diagram is presented and the performed modifications in the related codes are discussed.

Before focusing on the above mentioned steps, it is worth to present the workspace arranged for the project implementation.

4.1. Presentation of the workspace

This section presents the workspace in which the project has been developed that is located in GRCM lab of UPC north campus and it consists of the following components (see Figure 19):

- 2 Physical Machines (PM) with the Linux Ubuntu 14.4 distribution, used to install and run the OAI EPC (Physical Machine OAI EPC: PM OAI EPC) and the srsLTE eNB.
- 2 Virtual Machines (VM) with the Linux Ubuntu 14.4 distribution, used to install and run the OAI EPC (Virtual Machine OAI EPC: VM OAI EPC) and the 5G-EmPOWER Controller.
- 1 PC for programming the SIM cards, using the PySIM software.
- A USRP B210, as hardware platform for the SDR system. The USRP is in charge of providing connectivity between the users and the network. It has to be connected to the PC running the srsLTE eNB by means of a USB 3.0 cable. For the coupling between the USRP B210 and the antennas UTP cables have been used.
- Two pairs of antennas (Wi-Fi antennas) operating in band 7 and propagating at the frequency of 2600 MHz with MIMO configuration. One pair corresponds to the transmission antennas, while the other pair to the reception antennas.

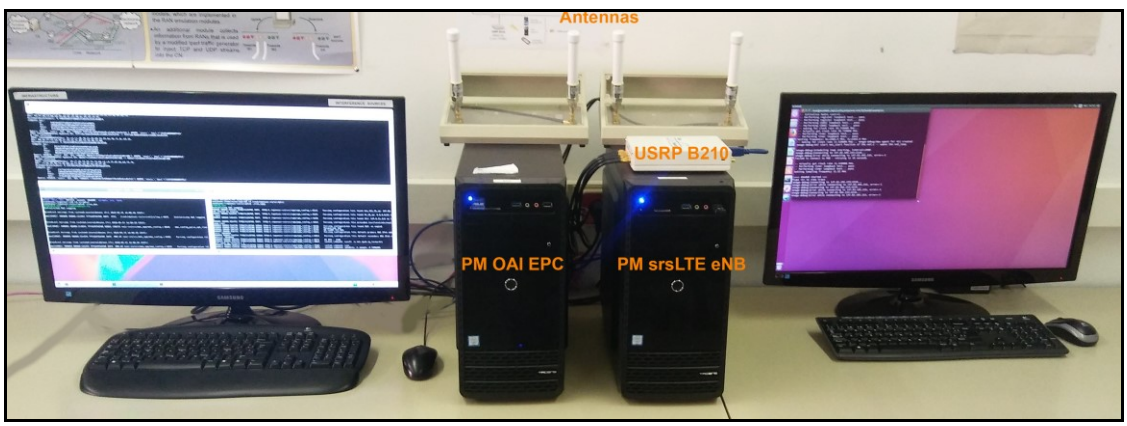


Figure 19. Project workspace in GRCM lab.

Apart from the above-mentioned items, User Equipment (UE: Mobile Equipment + SIM Card) is required to verify and test the network connectivity. The UEs used during this project are presented in the following (see Figure 20):

- Mobile Equipment (Cell phone or LTE Dongle): Samsung Galaxy SIII, Sony Xperia Z5, LG Nexus 5, and 2 LTE Dongles “Huawei E398”.
- The "sysmoUSIM-SJS1" SIM cards manufactured by sysmocom. The main advantage of these SIM cards is that they are re-programmable, so it is possible to edit any parameter in them even after the first time they were programmed. The related parameters to be programmed on these SIM Cards are set according to the network specifications to which each UE is intended to connect. Moreover, a SIM Card reader, such as the “Gemalto K30 USB token”, is used for the SIM programming.

Figure 20. User Equipments and SIM Card reader of GRCM lab used in this project.

A scheme of the project workspace in the GRCM lab is illustrated in Figure 21. In addition, the Figure depicts the functionality, the code that is running on, and the IP address of each machine. As mentioned before, while running, srsLTE eNB communicates with just one of the OAI EPCs, PM OAI EPC (red arrow) or VM OAI EPC (green arrow).

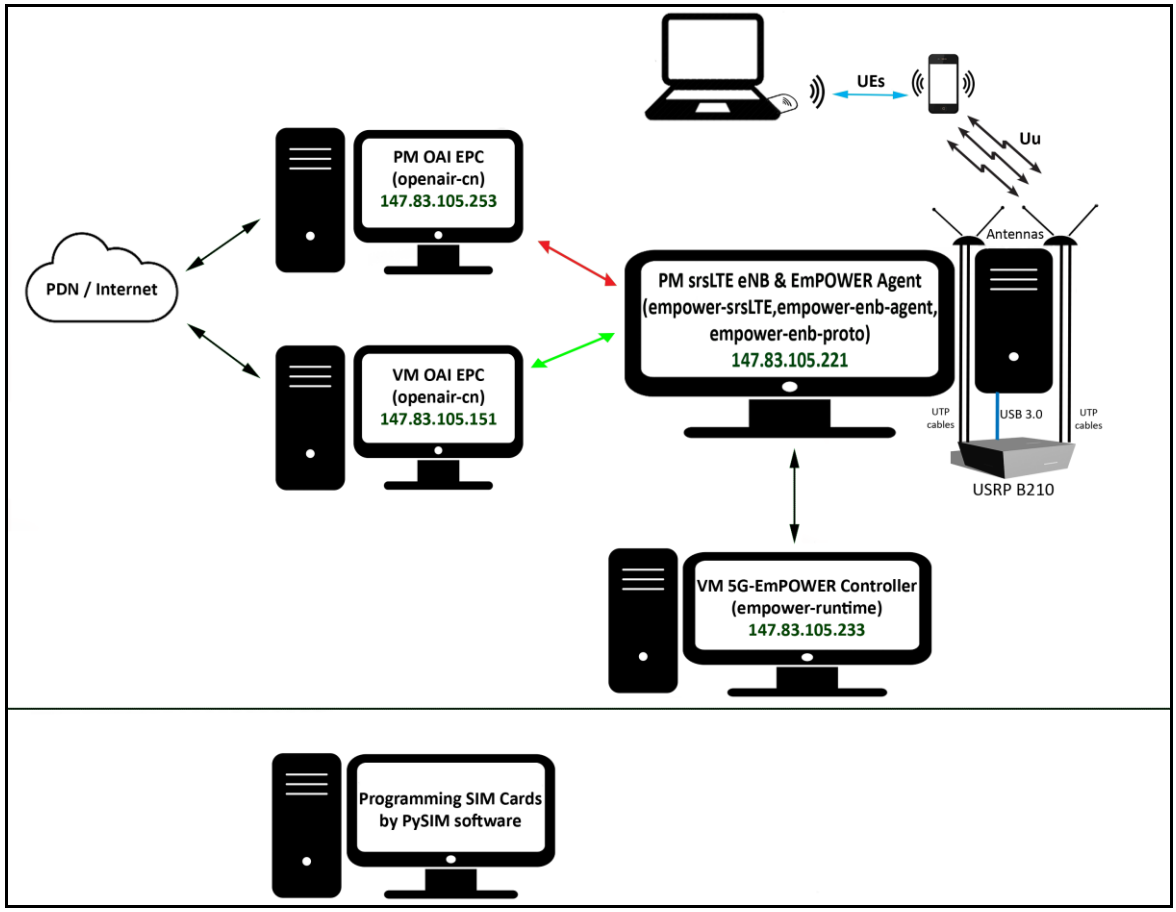


Figure 21. Scheme of the project workspace in GRCM lab.

4.2. OAI platform: EPC configuration and integration

As it can be seen from Figure 21, the core of LTE network, the OAI EPC, has been implemented in this project on both Physical and Virtual Machines. In the following, the Network architecture, General network parameters, and the configuration phase for both cases will be explained, however, let us notice that in the performed connectivity tests with different UEs has not presented difference in the results.

4.2.1. Network architectures (OAI EPC + srsLTE eNB)

In this section, the Network architecture of both implementations for the OAI EPC plus srsLTE eNB is depicted in Figures 22 and 23. The IP addresses allocation, interfaces` names, and port numbers are also included in the Figures. Notice that the interfaces used for communication inside the same host are defined as local addresses (lo), while the interfaces connecting the entities from different hosts (MME and SPGW with eNB) or the EPC entity (SPGW) with the external network are defined as host addresses (enp2s0, ens3). For simplicity, a summary table of IP addresses allocation for each case is included (see Table 2, 3).

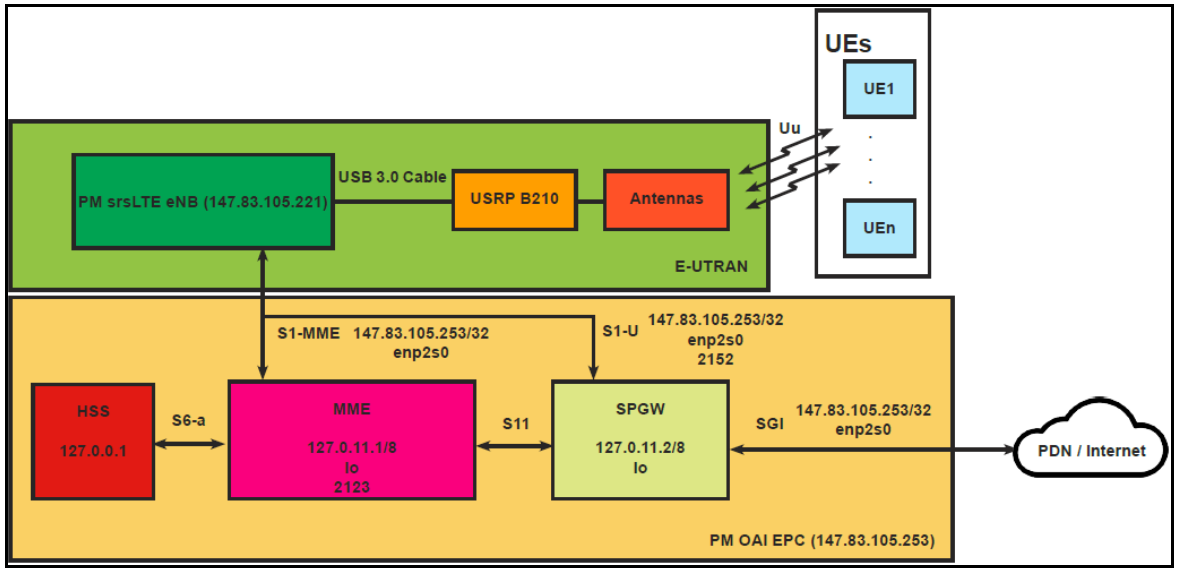


Figure 22. Network architecture of PM OAI EPC and PM srsLTE eNB configuration.

	eNB	MME	SPGW	PDN / Internet
eNB		147.83.105.221	147.83.105.221	
MME	147.83.105.253 (enp2s0)		127.0.11.1 (lo) 2123	
SPGW	147.83.105.253 (enp2s0) 2152	127.0.11.2 (lo)		147.83.105.253 (enp2s0)

Table 2. IP address allocation for PM OAI EPC and PM srsLTE eNB configuration.

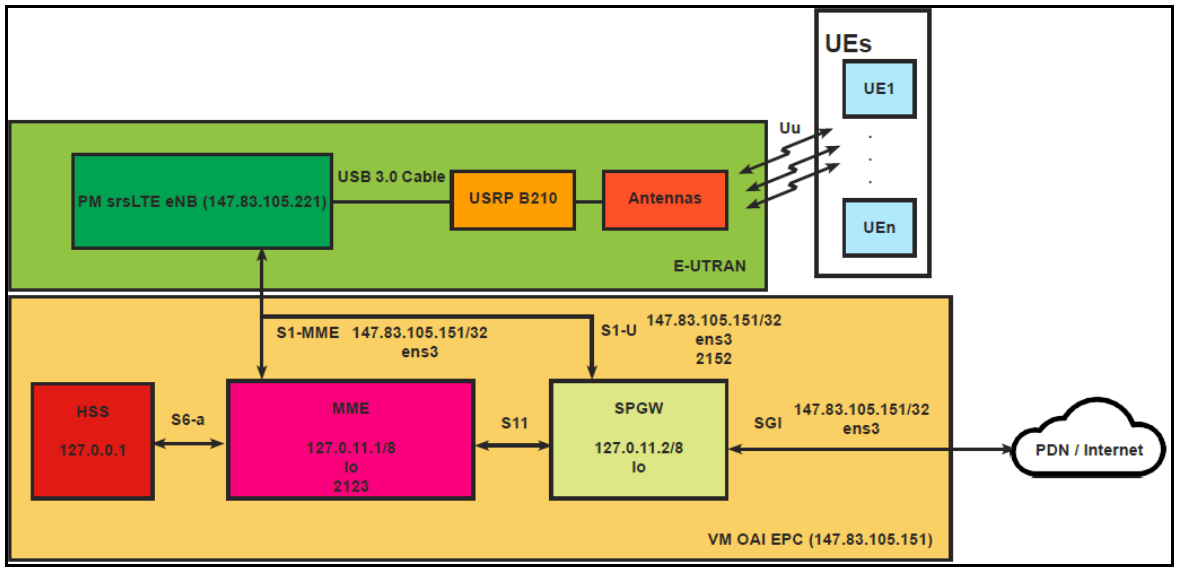


Figure 23. Network architecture of VM OAI EPC and PM srsLTE eNB configuration.

	eNB	MME	SPGW	PDN / Internet
eNB		147.83.105.221	147.83.105.221	
MME	147.83.105.151 (ens3)		127.0.11.1 (lo) 2123	
SPGW	147.83.105.151 (ens3) 2152	127.0.11.2 (lo)		147.83.105.151 (ens3)

Table 3. IP address allocation for VM OAI EPC and PM srsLTE eNB configuration.

4.2.2. General network parameters

As discussed before, some special parameters extracted from the radio network regulations have to be defined and then added in both the OAI EPC and srsLTE eNB configuration files. In the other words, a new mobile network operator has to be created for this project based on the valid combination of these parameters. In the following, these parameters are pointed out:

1. Public Land Mobile Network Identifier (PLMN Id) is a 5- or 6-digit parameter that identifies a specific mobile network operator. This identifier results from the Mobile Country Code (MCC) that determines the country to which the network belongs to, and the Mobile Network Code (MNC) that specifies the mobile network operator (**PLMN Id=MCCMNC**).
    - MCC is a 3-digit value assigned by the ITU. In this network, the MCC is 214 corresponding to the MCC of Spain.
    - MNC is a 2 or 3 digit assigned by the National Authority. The selection of MNC is carried out based on Table 4 that indicates the MNC of different network operators currently available in Spain [24]. As such, in order to create a new mobile network operator, a value that does not exist in the table must be selected.

MNC	Network Operator	MNC	Network Operator
01	Vodafone	17	R Cable y Telec. Galicia SA
03	Orange	18	Cableuropa SAU (ONO)
04	Yoigo	19	Simyo/KPN
05	Movistar	20	fon You Wireless SL
06	Vodafone Enabler España SL	21	Jazz Telecom SAU
07	Movistar	22	Digi Spain Telecom SL
08	Euskaltel SA	23	Lycamobile SL
09	Orange	25	Lycamobile SL
11	Orange	26	Lleida
15	BT España SAU	27	Truphone
16	Telecable de Asturias SA	32	ION Mobile

Table 4. Assignment of MNC to Network Operators in Spain.

Based on the MNC values shown in Table 4, the MNC selected for the network of this project is 91. Therefore, the corresponding PLMN Id for the deployed network in this project is **PLMN Id=21491**.

2. Globally Unique MME Identity (GUMMEI) is a unique MME identifier composed of the PLMN Id, the MME Group Identifier (MMEGI, unique within a PLMN Id) and the MME Code (MMEC). The last two parameters uniquely identify the MME within a particular network operator. In the case of this project, they have been set to the values: **MMEGI=4**, **MMEC=1**, So **GUMMEI=2149141**.
3. Tracking Area (TA) Identity List (TAI List) is the set of Tracking Area Identities (TAIs) that constitute the network. This parameter is composed of the PLMN Id and the Tracking Area Code (TAC), which identifies a tracking area within a particular network operator. It has to be mentioned that TAs are non-overlapping zones (see Figure 24) used to track the geographical location of the mobile devices that are in Idle mode [25]. In the network of this project, the TAI List is composed of **1 TA**, and so **TAC=1**.

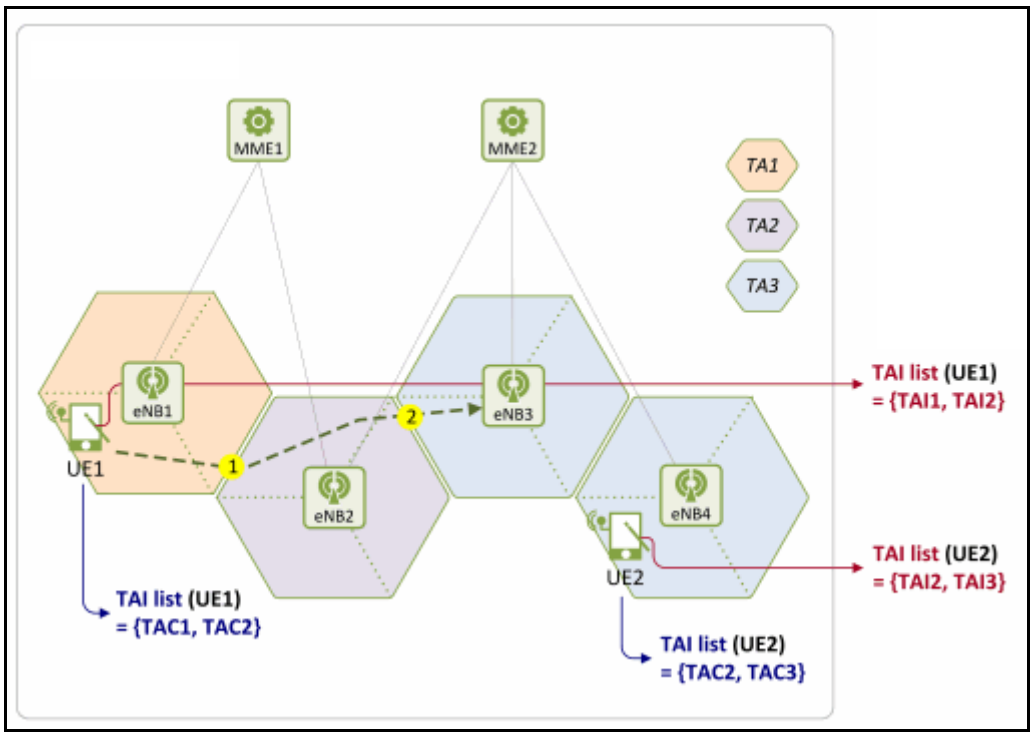


Figure 24. TA representation in geographical areas under network operator coverage.

Table 5 below, summarizes the general network parameters that have been defined for this project.

General network parameters		
MCC = 214	MNC = 91	PLMN Id = MCC+MNC = 21491
MMEGI = 4	MMEC = 1	GUMMEI= PLMN Id + MMEGI + MMEC = 2149141
TAC = 1	TAI List = PLMN Id + TAC = 214911	

Table 5. General network parameters for this project.

4.2.3. OAI EPC entities configuration

This section presents the general network parameters, interfaces, and IP addresses (defined previously) that have to be set for each of the OAI EPC entities, as well as the procedure that must be followed for the configuration of these parameters. A summary of the related configuration file is given in Table 6 below. Moreover, of configuration for both VM and PM versions of OAI EPC will be presented.

OAI EPC entities	Configuration files/tools				
HSS	SIM Cards` specifications network parameters	MYSQL Database	hss.conf	hss_fd.conf	acl.conf
MME	mme.conf	mme_fd.conf			
SPGW	spgw.conf				

Table 6. OAI EPC configuration files and tools.

The configuration process followed in order to configure the parameters for each entity in both VM and PM versions is explained in the next sections.

4.2.3.1. HSS configuration

HSS, in fact, operates as a database in the LTE network that stores the information of all the network users (SIM Cards). So first of all, the required information according to the manufacturing specifications of the SIM Cards and the general network parameters have to be classified and programmed on them. Then it is necessary to register the SIM Cards with this information in the database that is composed of MySQL tables, and finally based on the database, the parameters in the HSS configuration files have to be set.

- **SIM Cards programming according to their specifications and general network parameters (common for PM OAI EPC and VM OAI EPC)**

  All the required SIM Cards` specifications are as follows:

  - IMSI: International Mobile Subscriber Identity. The IMSI has a maximum of 15 digits and is the combination of MCC | MNC | MSIN. The last parameter is the Mobile Subscriber Identification Number and it is a nine- or ten-digit code that identifies the Mobile Station.
  - Key: This parameter and the OPc are the ones used for authenticating the users in a mobile network. The Key or K is a 128-bit master key located in the USIM and the HSS entity by the carrier.
  - OP Key: It is a 32-bit Operator Variant Algorithm Configuration Field. Each operator must have a different OP Key value. From the OP Key and the SIM Card Key, the OPc will be computed. The related value has been provided by OAI community [30].
  - CC: Country Code, in Spain it is 34.

Table 7, summarizes 4 SIM Cards` manufacturing specifications and the corresponding network parameters:

SIM Cards	number 1	number 2	number 3	number 4
MCC	214			
MNC	91			
TAI	1			
IMSI	214910000009911	214910000009914	214910000009915	214910000009916
SIM Card Key	29C7D674257AF3B0 C6C9F7879663F5FB	4403368D7F3ABEE7 CBED3155D134ABFA	3EC6CED8F8115403 54558 A9BF18C631B	3AFA3506B8F4775F BFA808F56F59D801
OP Key	11111111111111111111111111111111			
CC	34			

Table 7. SIM Cards` specifications and network parameters.

As such, the SIM Cards are programmed based on the above presented parameters and then they are registered in the database tables.

- **Database configuration (common for PM OAI EPC and VM OAI EPC)**

The database can be accessed using a Linux terminal, or a web-based user interface called "phpMyAdmin" (see Figure 25). Let us notice that the web interface facilitates the access to the database, however the introduction of some parameters is easier to be carried out using the terminal. phpMyAdmin can be accessed through:

http://<localhost IP address of OAI EPC>/phpmyadmin/

which in the case of this project is:

- PM OAI EPC: http://<127.0.0.1>/phpmyadmin/
- VM OAI EPC: http://<147.83.105.151>/phpmyadmin/

Figure 25. phpMyAdmin user interface of the database.

A detailed description of the procedure followed for the modification of the database through phpMyAdmin interface is included in Annex [1].

The MySQL database name used in this project is: oai_db. It includes different tables, such as apn, mmeidentity, pdn, pgw, terminal-info and users. SIM Cards information has to be registered in:

- pdn: It contains the association between a network subscriber (user_imsi) and an Access Point Name (APN) and its Quality of Service (QoS) parameters.
- mmeidentity: It contains the record corresponding to the MME of OAI EPC. In this project just one MME has been implemented, so a single value has to be defined for the id section in this table.
- users: It contains the information about all the subscribers of the network (USIM card).

The information of Table 7 and the default values provided by the OAI community have to be set in these three tables as illustrated in the following Figures (26, 27, 28):

id	apn	pdn_type	pdn_ipv4	pdn_ipv6	aggregate_ambr_ul	aggregate_ambr_dl	pgw_id	users_imsi	qci	priority_level	pre_emp_cap	pre_emp_vul	LIPA-Permissions
1	oai.ipv4	IPv4	0.0.0.0	0:0:0:0:0:0	50000000	100000000	3	214910000009914	7	7	DISABLED	ENABLED	LIPA-only
2	oai.ipv4	IPv4	0.0.0.0	0:0:0:0:0:0	50000000	100000000	3	214910000009911	7	7	DISABLED	ENABLED	LIPA-only
3	oai.ipv4	IPv4	0.0.0.0	0:0:0:0:0:0	50000000	100000000	3	214910000009915	7	7	DISABLED	ENABLED	LIPA-only
4	oai.ipv4	IPv4	0.0.0.0	0:0:0:0:0:0	50000000	100000000	3	214910000009916	7	7	DISABLED	ENABLED	LIPA-only

Figure 26. pdn table of the oai_db database.

idmmeidentity	mmehost	mmerealm	UE-Reachability Indicates whether the MME supports UE Reachability Notifcation
1	nano.openair4G.eur	openair4G.eur	0

Figure 27. mmeidentity table of the oai_db database.

imsi IMSI is the main reference key.	msisdn The basic MSISDN of the UE (Presence of MSISDN is ...	imei International Mobile Equipment Identity	imei_sv International Mobile Equipment Identity Software V...	ms_ps_status Indicates that ESM and EMM status are purged from ...	rau_tau_timer	ue_ambr_ul The Maximum Aggregated uplink MBRs to be shared ac...	ue_ambr_dl The Maximum Aggregated downlink MBRs to be shared ...	access_restriction Indicates the access restriction subscription info...	mme_cap Indicates the capabilities of the MME with respect...	mmeidentity_idmmeidentity
214910000009915	34600000003	NULL	NULL	NOT_PURGED	120	50000000	100000000	47	0000000000	1
214910000009914	34600000001	NULL	NULL	NOT_PURGED	120	50000000	100000000	47	0000000000	1
214910000009911	34600000002	NULL	NULL	NOT_PURGED	120	50000000	100000000	47	0000000000	1
214120000009916	34600000005	NULL	NULL	NOT_PURGED	120	50000000	100000000	47	0000000000	1

key UE security key	RFSP-Index An index to specific RRM configuration in the E-UT...	urrp_mme UE Reachability Request Parameter indicating that ...	sqn	rand	OPc Can be computed by HSS
3ec6ced8f811540354558a9bf18c631b	1	0	00000000000000011264	0119ff30915c90277c5cb92f99744f2c	001bffa7e850af29a5fe03725538f200
4403368d7f3abee7cbed3155d134abfa	1	0	00000000000000031616	a7120c4cb5532e16579d60579aaabe1b	977fe0c486b344668ce5dd4c54e9a754
29c7d674257af3b0c6c9f7879663f5fb	1	0	00000000000000015360	048e4d1a60000954929fc840867bd212	cf66354bf1d20da6ff0c61816ac0afa5
ee3701e7bdf200c42207a073891f8202	1	0	00000000000000007552	0682bf14d104a5568a26b007c39a9779	a469f2da4cadeaf3d51d387b1c33dbd6

Figure 28. users table of the oai_db database.

- **hss.conf (common for PM OAI EPC and VM OAI EPC)**

  This configuration file is used to define the location and access data of the database. Moreover, the OPERATOR key parameter (OP key) has to be included in this file. All users of the same network operator must have the same OP Key parameter in their SIM Cards` configurations. An example of the hss.conf file used in this project is shown in Figure 29:

hss.conf

```
#########################################################################################
HSS :
{
## MySQL mandatory options
MYSQL_server = "127.0.0.1";     # HSS S6a bind address
MYSQL_user   = "root";  # Database server login
MYSQL_pass   = "epc";  # Database server password
MYSQL_db     = "oai_db";        # Your database name

## HSS options
#OPERATOR_key = "1006020f0a478bf6b699f15c062e42b3"; # OP key matching your database
OPERATOR_key = "11111111111111111111111111111111"; # OP key matching your database

RANDOM = "true";

## Freediameter options
FD_conf = "/usr/local/etc/oai/freeDiameter/hss_fd.conf";
};
```

Figure 29. Configuration of hss.conf file.

- **hss_fd.conf, acl.conf (common for PM OAI EPC and VM OAI EPC)**

  According to OAI community [16], these two files will be created and configured during the compilation process based on the host parameters (hostname, realm...) where the HSS has been implemented. In Figures 30 and 31 the content of these two files is presented:

hss_fd.conf

```
# -------- Local ---------
# The first parameter in this section is Identity, which will be used to
# identify this peer in the Diameter network. The Diameter protocol mandates
# that the Identity used is a valid FQDN for the peer. This parameter can be
# omitted, in that case the framework will attempt to use system default value
# (as returned by hostname --fqdn).
Identity = "hss.openair4G.eur";

# In Diameter, all peers also belong to a Realm. If the realm is not specified,
# the framework uses the part of the Identity after the first dot.
Realm = "openair4G.eur";


# Specify the addresses on which to bind the listening server. This must be
# specified if the framework is unable to auto-detect these addresses, or if the
# auto-detected values are incorrect. Note that the list of addresses is sent
# in CER or CEA message, so one should pay attention to this parameter if some
# adresses should be kept hidden.
#ListenOn = "127.0.0.1";
```

Figure 30. Configuration of hss_fd.conf file.

acl.conf

```
# ALLOW_OLD_TLS : we accept unprotected CER/CEA exchange with Inband-Security-Id = TLS
# ALLOW_IPSEC   : we accept implicitly protected connection with with peer (Inband-Security-Id = IPSec)
# It is specified for example as:
# ALLOW_IPSEC vpn.example.net vpn2.example.net *.vpn.example.net

ALLOW_OLD_TLS   *.openair4G.eur
```

Figure 31. Configuration of acl.conf file.

4.2.3.2. MME configuration

MME operates as the main entity of the control plane in the network. It includes two configuration files that have to be adjusted according to the parameters defined in the previous sections.

- **mme.conf**

  The network parameters (GUMMEI, TAI List), IP addresses, and interfaces through which the MME communicates with the other entities of the network (SPGW, srsLTE eNB) are configured in this file, as shown in Figures 32 and 33. There are some parameters that must be defined differently based on whether a PM or VM OAI EPC is used.

```
mme.conf

# ------- MME served GUMMEIs
# MME code DEFAULT  size = 8 bits
# MME GROUP ID size = 16 bits
GUMMEI_LIST = (
     {MCC="214" ; MNC="91"; MME_GID="4" ; MME_CODE="1"; }
);

# ------- MME served TAIs
# TA (mcc.mnc:tracking area code) DEFAULT = 208.34:1
# max values = 999.999:65535
# maximum of 16 TAIs, comma separated
# !!! Actually use only one PLMN
TAI_LIST = (
     {MCC="214" ; MNC="91";  TAC = "1"; }
);
```

Figure 32. Configuration of GUMMEI and TAI List in mme.conf file (common for PM OAI EPC and VM OAI EPC).

```
mme.conf

NETWORK_INTERFACES :
{
    # MME binded interface for S1-C or S1-MME communication (S1AP)

    MME_INTERFACE_NAME_FOR_S1_MME         = "enp2s0";
    MME_IPV4_ADDRESS_FOR_S1_MME           = "147.83.105.253/32";   <-- PM OAI EPC

   #MME_INTERFACE_NAME_FOR_S1_MME         = "ens3";
   #MME_IPV4_ADDRESS_FOR_S1_MME           = "147.83.105.151/32";   <-- VM OAI EPC

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

Figure 33. Configuration of network interfaces in mme.conf file (for PM OAI EPC and VM OAI EPC).

- **mme_fd.conf (common for PM OAI EPC and VM OAI EPC)**

  This file contains the Identity parameter that indicates how the MME is identified by the name of the host machine where the MME is implemented. In addition, the Realm parameter is included, which refers to the domain name of that host. These two are extracted from the mmeidentity table of the oai_db (see Figure 34).

```
mme_fd.conf

# -------- Local ---------
Identity = "nano.openair4G.eur";
Realm = "openair4G.eur";
```

Figure 34. Configuration of mme_fd.conf file.

4.2.3.3. SPGW configuration

The SPGW operates as the unit in charge of the data plane in the LTE network. There is only one single SPGW configuration file (spgw.conf) split into two sections. One section corresponds to the S-GW configuration, while the other to the P-GW. The DNS and MTU values for the UEs are configured in the P-GW section. Let us notice, that also in this case there are some differences in the configuration depending on whether a PM or VM OAI EPC is used. A configuration file example is shown in Figures 35 and 36.

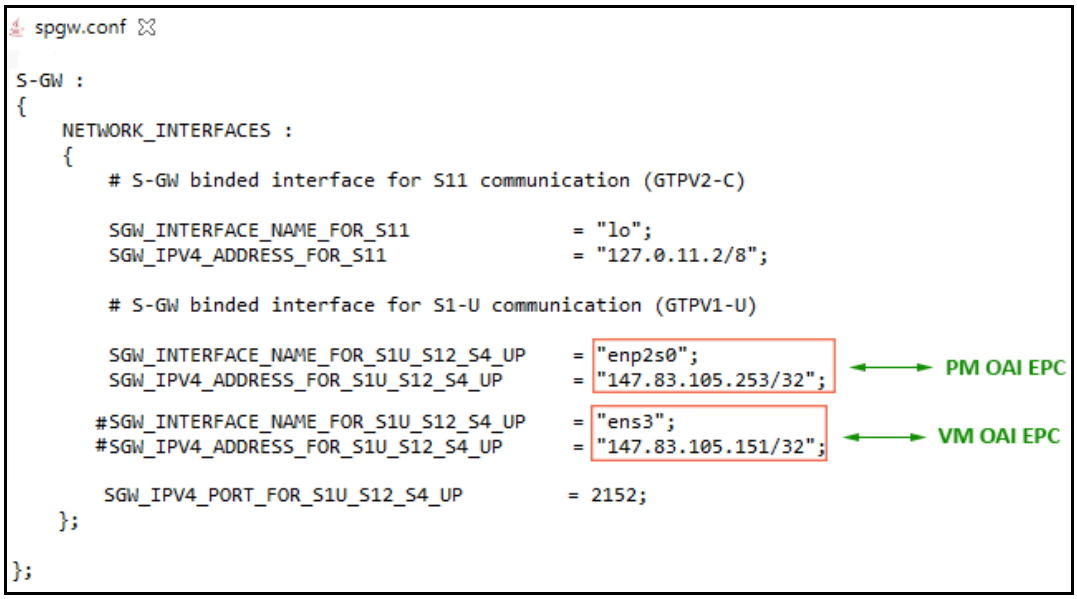


Figure 35. Configuration of S-GW section of spgw.conf file (for PM OAI EPC and VM OAI EPC).

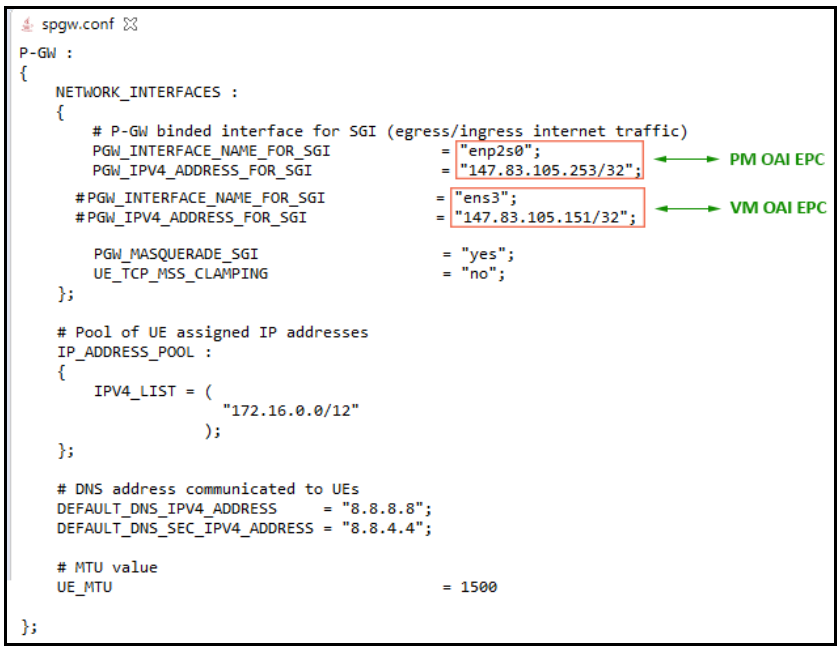


Figure 36. Configuration of P-GW section of spgw.conf file (for PM OAI EPC and VM OAI EPC).

This part finalizes the OAI EPC configuration of this project.

4.3. srsLTE platform: eNB configuration and integration

As discussed before srsLTE is one of the main three platforms used in this project. Specifically, srsLTE eNB plays the role of E-UTRAN in the current network (implemented on a Physical Machine) and interacts with the OAI EPC (VM or PM). Since the network architecture and general network parameters have been defined and explained in previous sections, just the configuration phase will be presented in the following.

4.3.1. srsLTE eNB entity configuration

The configuration of some special characteristics related to the srsLTE eNB itself (eNB, cell, and physical cell identifiers) and IP addresses of the entities with which it communicates are presented in this section. In the case of the 5G-EmPOWER controller, its port number has also to be included. All of these parameters have to be added in the single srsLTE eNB configuration file (enb.conf) as displayed in Figure 37:

enb.conf

```
#####################################################################
                    srsENB configuration file

# eNB configuration
[enb]
enb_id = 0x19B                  # 20-bit eNB identifier.
cell_id = 0x01                  # 8-bit cell identifier.
phy_cell_id = 1
tac = 0x0001                    # 16-bit Tracking Area Code.
mcc = 214                       # Mobile Country Code
mnc = 91                        # Mobile Network Code
mme_addr = 147.83.105.253       # IP address of MME for S1 connnection <--> PM OAI EPC IP address
#mme_addr = 147.83.105.151      # IP address of MME for S1 connnection <--> VM OAI EPC IP address
ctrl_addr = 147.83.105.233      # IP address of 5G-EmPOWER Controller
ctrl_port = 2210                # TCP port of 5G-EmPOWER Controller
gtp_bind_addr = 147.83.105.221 # Local IP address to bind for GTP connection <--> PM srsLTE eNB IP address
n_prb = 50                      # Number of Physical Resource Blocks (6,15,25,50,75,100)
```

Figure 37. Configuration of enb.conf file.

4.4. Launching the network

Once all the corresponding elements are configured, the network can be started up by launching the different entities. Notice that the execution process of these entities has to be performed in a specific order. In particular, the HSS has to be launched first, then the MME, the SPGW and last the srsLTE eNB. The next Figures depict all the terminals corresponding to the different entities and display the most important messages in each of them appearing during the running process. Since there is no difference in the appeared messages in Linux terminals for the case of PM or VM OAI EPC (except just the IP addresses), only the VM OAI EPC case has been included in the following Figures. Notice that some brief explanations have been added for all of the cases (see Figure 38 until 41).

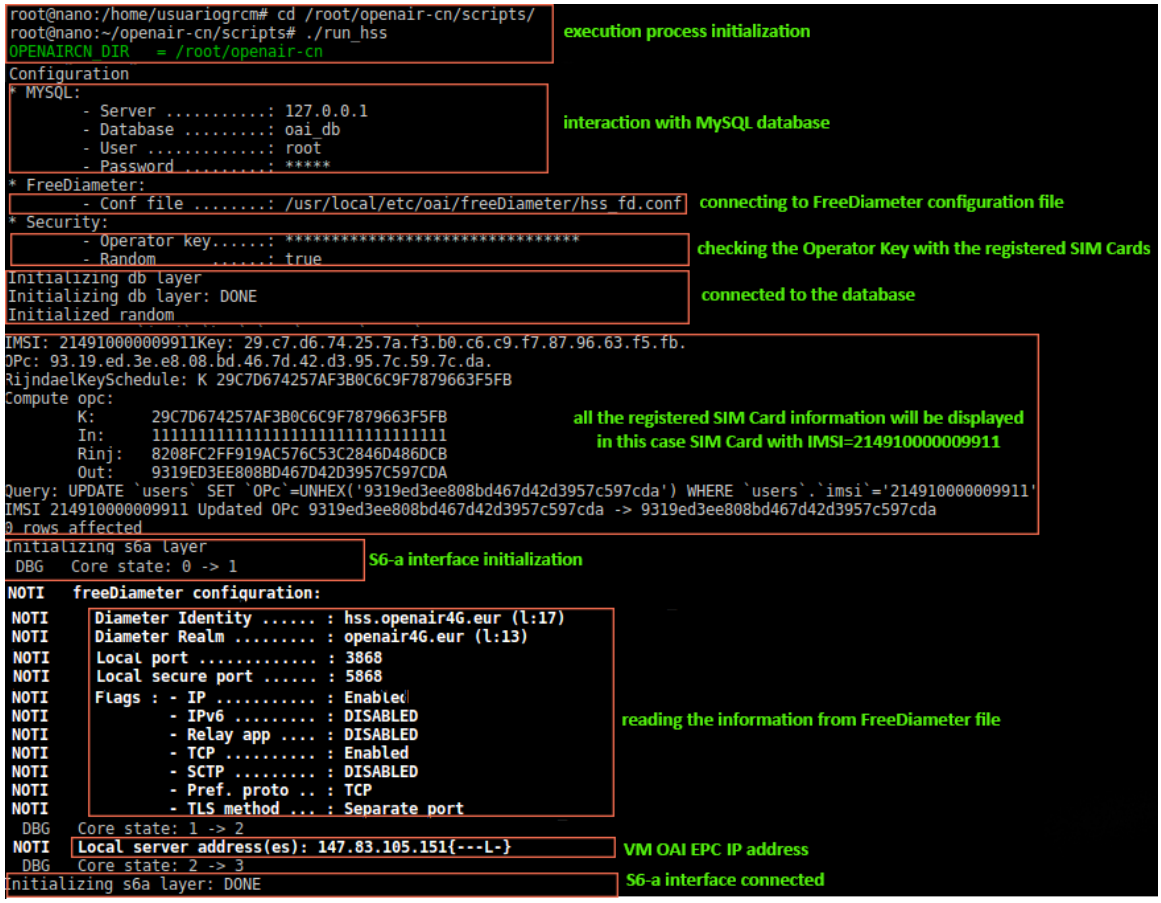


Figure 38. HSS entity Linux terminal.

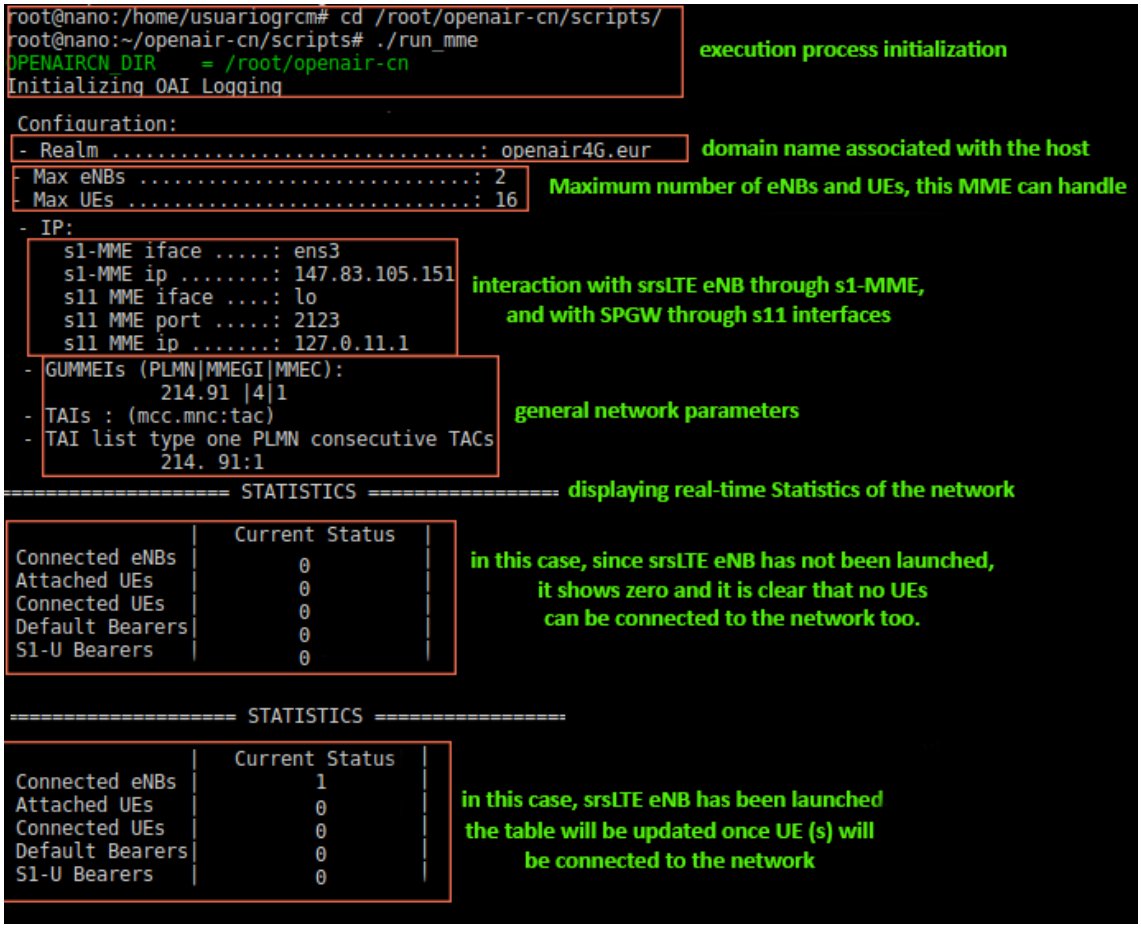


Figure 39. MME entity Linux terminal.

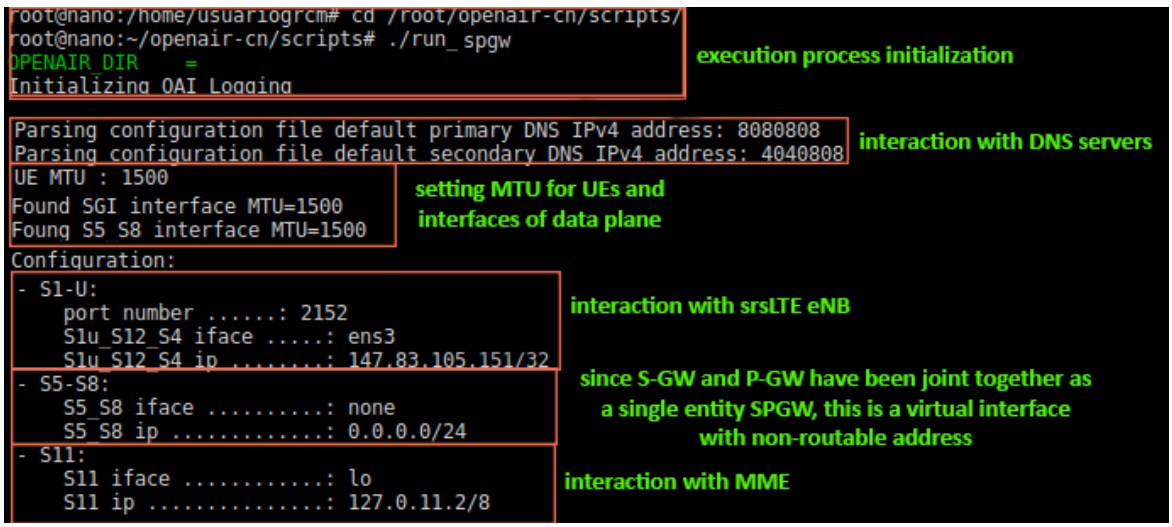


Figure 40. SPGW entity Linux terminal.

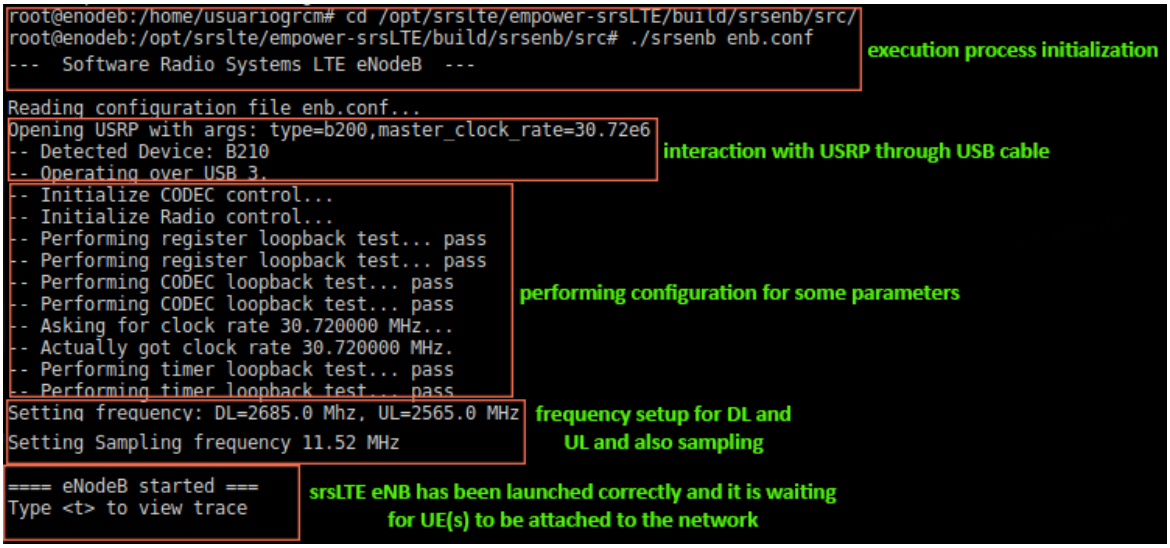


Figure 41. srsLTE eNB entity Linux terminal.

4.5. UE connectivity tests

After launching the network, some connectivity tests have to be performed in order to verify the correct operation of the network and that UEs can access the internet services. To do so, initially of all the UEs (cell phones and dongles) have to be configured. The Access Point Name (APN) parameter (previously defined in the pdn table of the database as "oai.ipv4") has to be introduced in all the UEs that want to connect to the network (see Annex [1]). After that, UEs will be able to be attached and then connected to the created network. Notice that the created mobile network now will be recognized by the corresponding values previously defined for MCC=214 and MNC=91. In the following, the network accessibility by UEs will be verified.

- **In the case of cell phones**
    - RFBENCHMARK is an application capable of detecting existing mobile networks in a geographical area. For this reason, it has been installed on the cell phones for connectivity tests in this project. RFBENCHMARK indicates the network in which the cell phone has been connected to and also detects other reachable mobile network operators in a specific range of where the cell phone is located. The application indicates the best mobile operator among the detected networks according to the best signal quality. Figure 42, shows that by using RFBENCHMARK, the created network has been detected and it is the best mobile operator available inside the testing region. So, this confirms that the network has been well implemented.

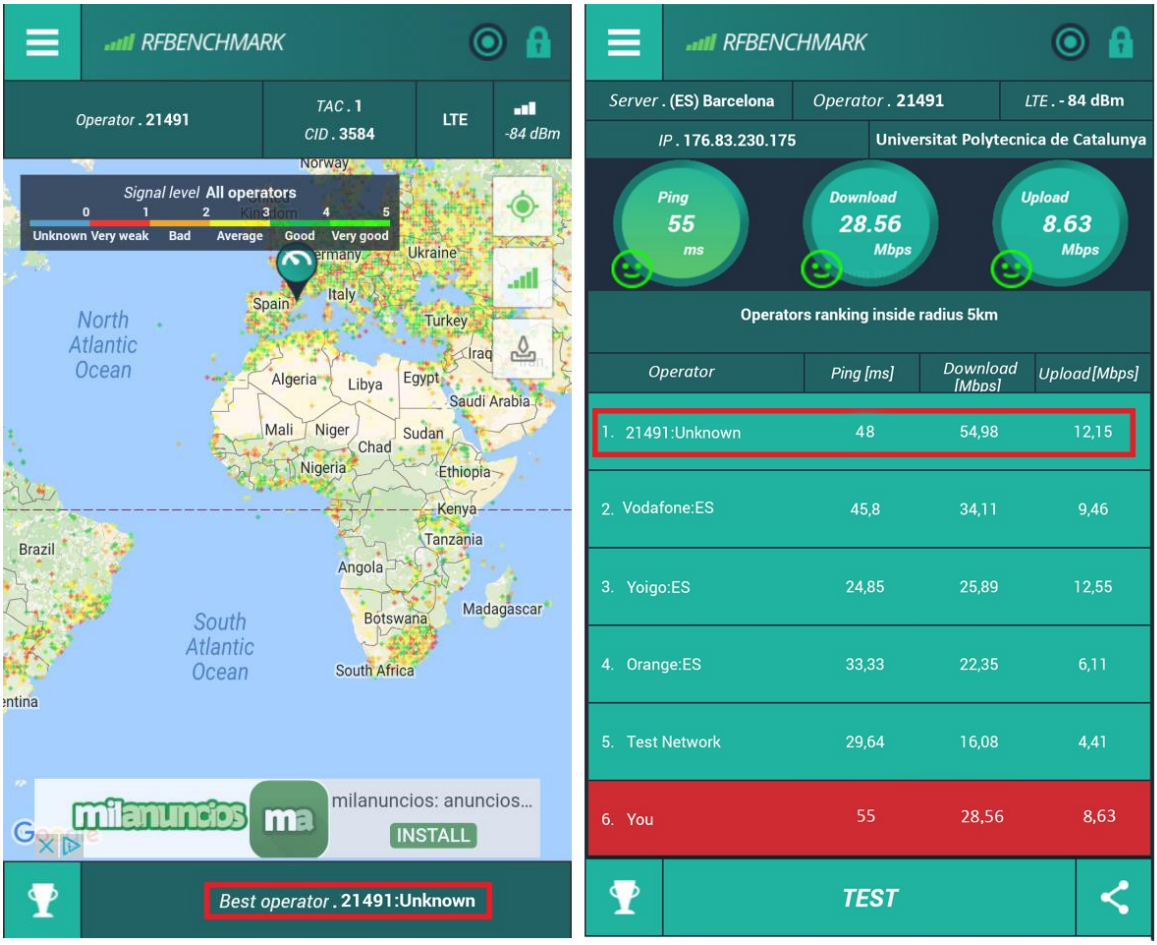


Figure 42. RFBENCHMARK detects the created network and displays its quality.

- Qualipoc from SwissQual, is another application that provides more detailed information about the network, such as radio signal quality parameters, signal power, eNB cell information, communications technology etc. Figure 43, illustrates that the created network is reachable with appropriate signal power.

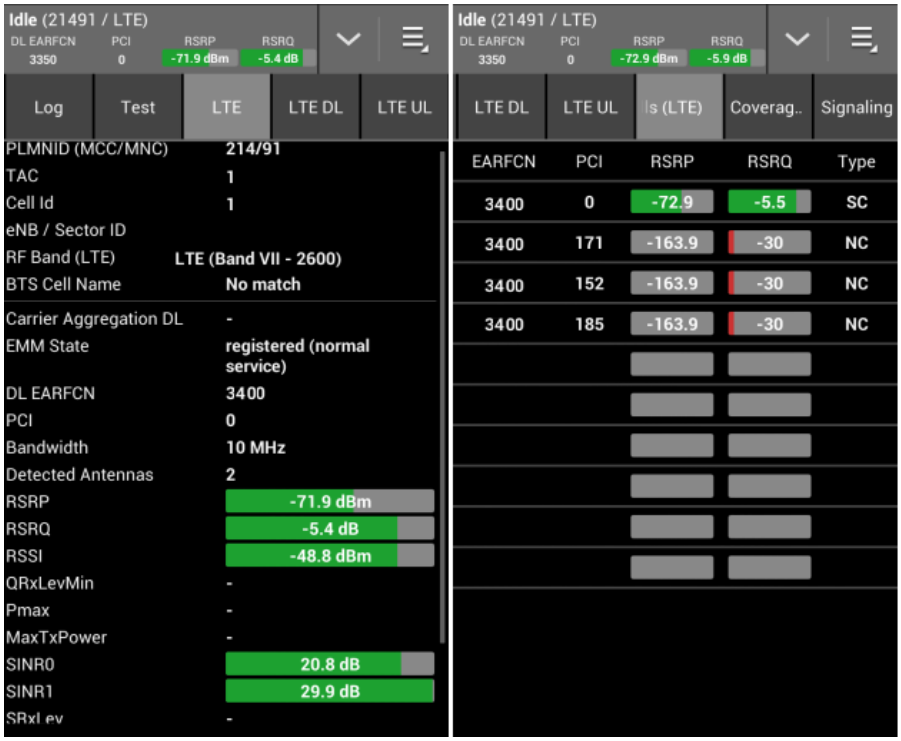
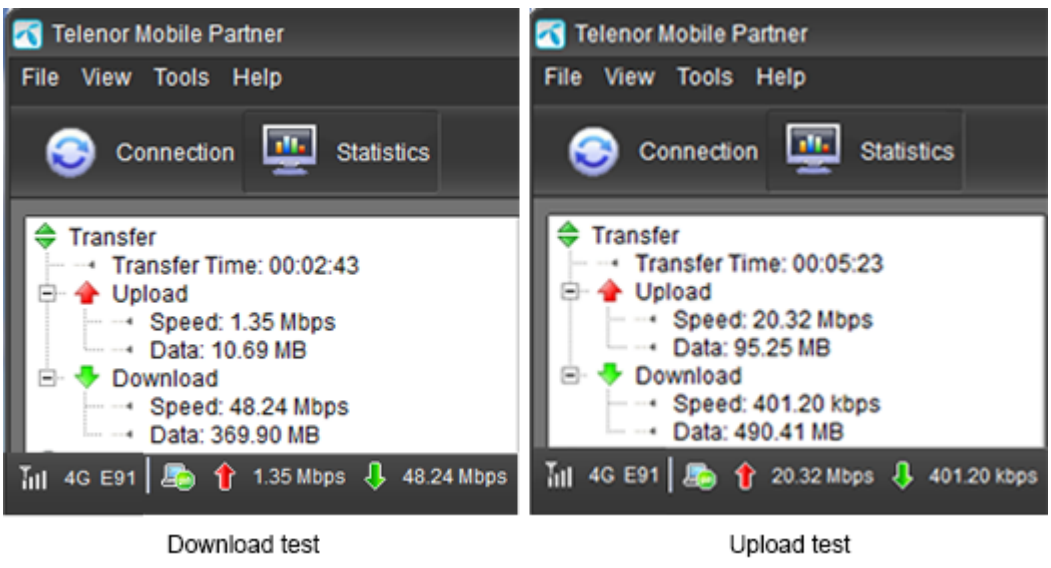


Figure 43. Qualipoc detects the created network and displays its quality.

- **In the case of dongles**

  - Telenor Mobile Partner is the software provided by the manufacturer, which has to be installed on the laptop that wants to connect to the network. This software has a Statistics tab visualizing the network characteristics. Figure 44, shows the information that appears in this tab when doing some tests in order to measure the performance of the network for downloading and uploading large amount of data.



Download test Upload test

Figure 44. Telenor Mobile Partner Software Statistics tab.

Up to now, the network information that appears on the UEs has been analyzed. Next, the information which appearing in the Linux terminals of the network entities` are pointed out.

While a UE is attaching to the network, the information of its SIM Card will be displayed in the HSS terminal and after the authentication process, this UE (SIM Card) can connect to the network (see Figure 45):

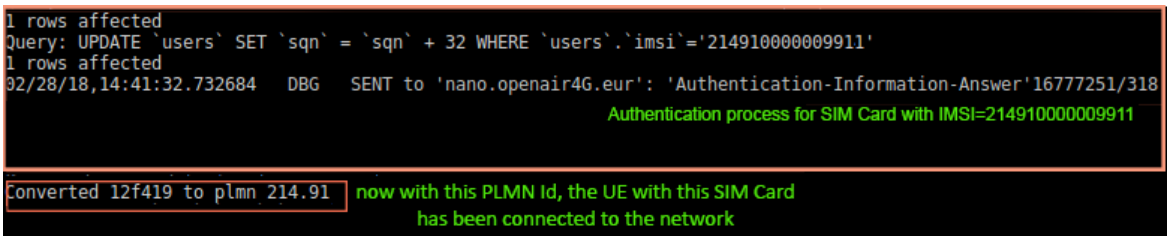


Figure 45. HSS terminal while a UE attachment.

Once the UE has been attached and connected to the network, the MME terminal also updates it statistics table as depicted in Figure 46:

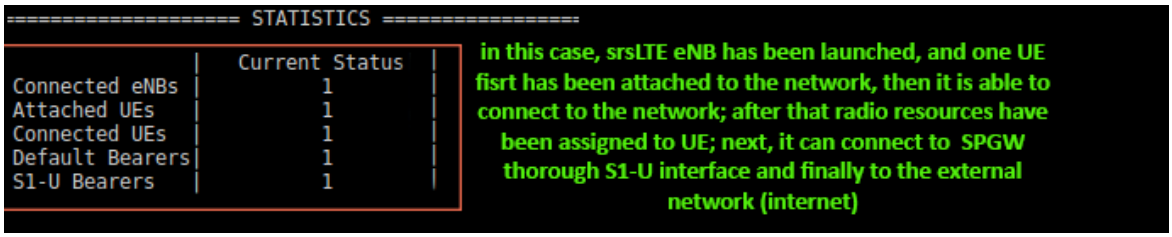


Figure 46. MME terminal once the UE has been attached to the network.

Next, a Radio Network Temporary Identifier (RNTI) value is assigned to the UE by the network. An RNTI value is used to identify the information dedicated to a particular subscriber on the radio interface. Finally, with the radio resources provided through the srsLTE eNB, the UE can connect to the internet. Apart from the RNTI value, some physical parameters can be also seen in the eNB terminal for both DL and UL, such as Modulation and Coding Schemes (mcs), Bit rate(brate) and SNR as depicted in Figure 47:

```
RACH:  tti=2691, preamble=25, offset=4, temp_crnti=0x46
User 0x46 connected
```

assigning rnti value (0x46) to UE when it has been connected

DL							UL			
rnti	cqi	ri	mcs	brate	bler	snr	phr	mcs	brate	bler	bsr
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0
46	15	0.00	-nan	0.0	0%	-nan	0.0	-nan	0.0	0%	0.0

physical parameters of UE in both DL and UL

Figure 47. Tracing messages in srsLTE eNB terminal after UE connection.

4.6. 5G-EmPOWER platform: Configuration and integration

Until now, the Core and E-UTRAN parts of the LTE network have been implemented by using the OAI EPC and srsLTE eNB platforms respectively. As discussed before separating the control plane and data plane as one of the main trends in future networks has been pursued in recent years based on SDN-NFV technology. 5G-EmPOWER controller, as one of the SDN-NFV based implemented software, is a multi-access computing operating system. It supports lightweight virtualization and heterogeneous radio access technologies [20].

5G-EmPOWER is responsible for managing multiple slices or tenants over a common physical infrastructure and, consequently, their radio resources. In Figure 48, an overview scheme of this system, including the involved interactive elements is depicted:

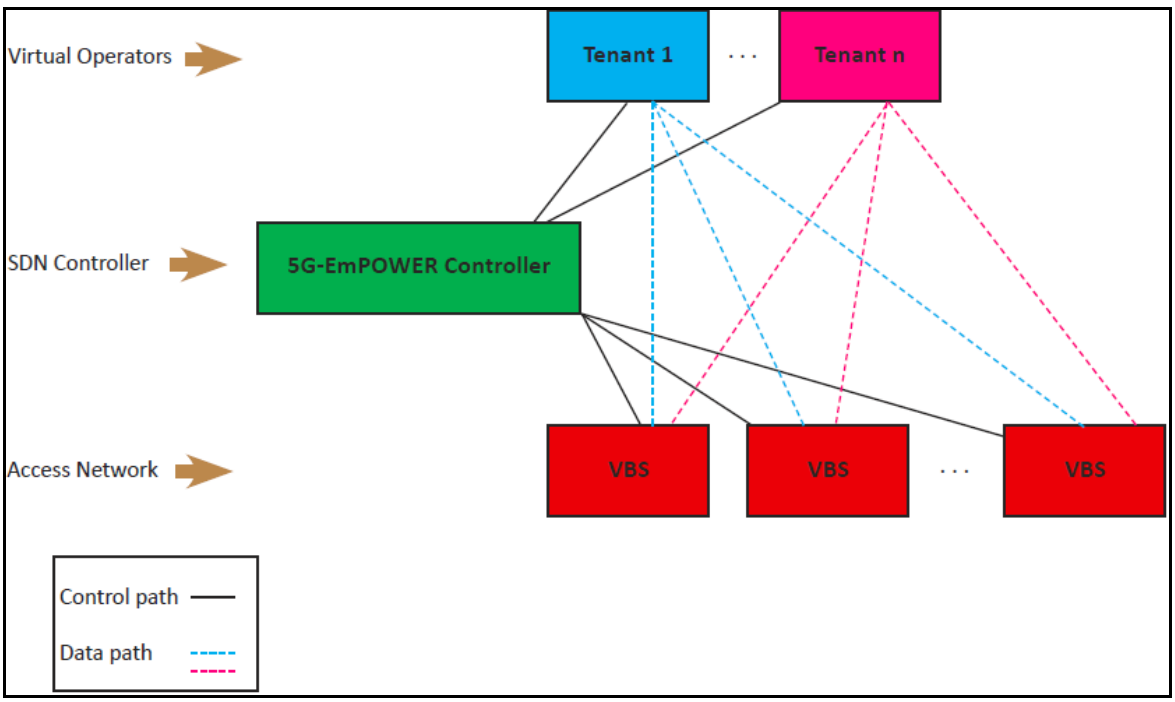


Figure 48. 5G-EmPOWER scheme.

In this Figure:

- The Tenant or Mobile Virtual Network Operator (MVNO) provides services to the network users associated to that specific tenant. They have a guaranteed Service Level Agreement (SLA) in the network.
- The 5G-EmPOWER Controller is responsible for managing the different tenants and their resources among the different VBSs assigned to them and it guarantees that the agreed SLA is accomplished.
- Virtual Base Stations (VBSs) are he LTE networks that allow the users to connect to the network and use different services.

4.6.1. Network architecture (5G-EmPOWER + (srsLTE eNB & OAI EPC))

In this section, the Network architecture of the 5G-EmPOWER controller along with the previous integrated platforms is depicted in Figure 49. The IP address and port number corresponding to the controller are also included in the Figure (since there is no difference of using VM or PM versions of OAI EPC, only the PM version has been included).

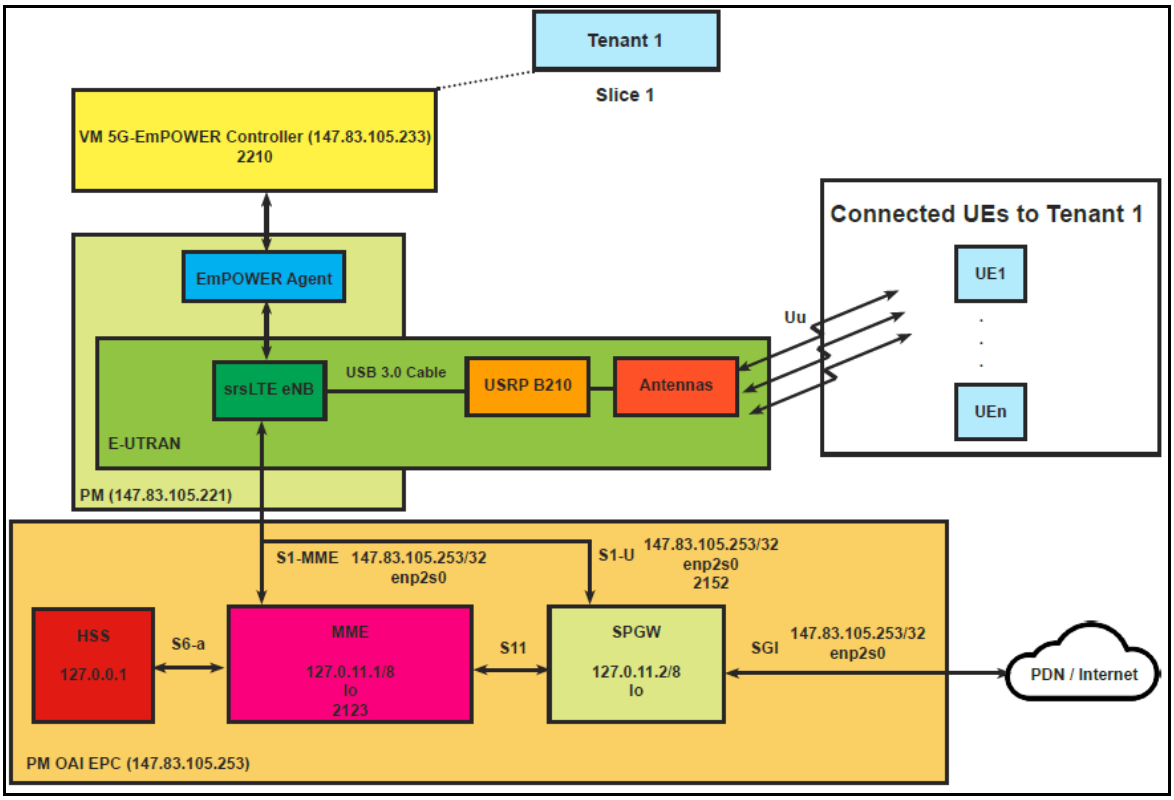


Figure 49. Network architecture of VM 5G-EmPOWER controller with previous integrated platforms.

Notice that it is not mandatory to implement the 5G-EmPOWER controller in a Virtual Machine. It is just the implementation that has been used in the project. Implementing various rules of assigning radio resources to different users corresponding to this tenant

has a great advantage in radio resources management which is performed by 5G-EmPOWER in the network.

In the following sections, the way to access the 5G-EmPOWER through its interface is initially presented. Next, the role of the EmPOWER Agent entity is analyzed. Then the agent code section of the empower-srsLTE is configured and activated and finally, the srsLTE eNB (VBS) is integrated with the 5G-EmPOWER through the controller interface.

4.6.2. Presentation of 5G-EmPOWER interface

The 5G-EmPOWER Controller, in order to facilitate the access to it, provides a web-based interface with high level programmability aspect that enables the tenants' creation and also the association with VBS or other wireless termination points, such as Wi-Fi Access Points. empower-runtime is the software code which has to be installed on the corresponding machine (VM in this project) in order to integrate the 5G-EmPOWER controller with all of its libraries. Figure 50, displays the Linux terminal of the 5G-Empower controller with a summary of the main initial messages appearing while running it:

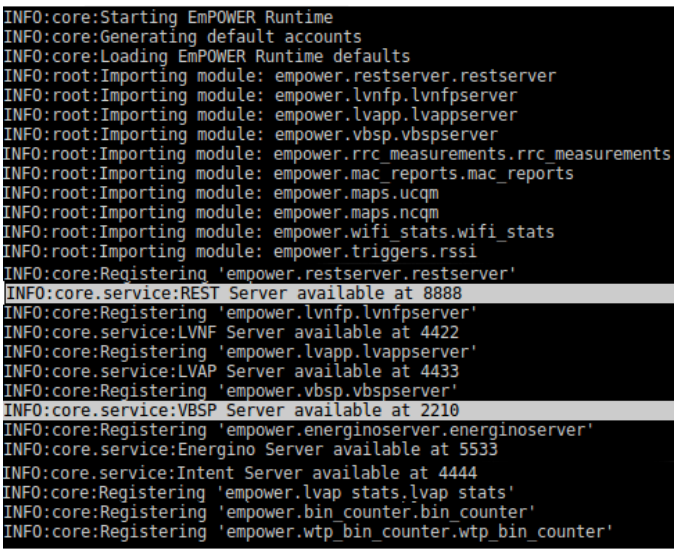


Figure 50. The Linux terminal of 5G-EmPOWER while running it.

Notice that, in Figure 50, the highlighted lines correspond to the:

- Port number used to access the 5G-EmPOWER controller web interface (8888).
- Port number used for the interaction between the 5G-EmPOWER controller with the srsLTE eNB (VBS) through the EmPOWER Agent. This is the port number previously added in the srsLTE eNB configuration file (2210).

So in general, the web interface can be accessed through:

http://<IP address of 5G-EmPOWER controller>:8888/

which in this project is:

http://147.83.105.233:8888/

Therefore, the 5G-EmPOWER web interface appears in the browser as depicted in Figure 51:

Figure 51. 5G-EmPOWER web interface.

5G-EmPOWER has two types of access, which are the administrator and regular user accesses.

- In the case of administrator access (with username/password: root):

  It is composed of different tabs belonging to either LTE or Wi-Fi technologies. The tabs that are involved in the LTE network are the following:

  - **Tenants:** corresponds to the active tenants which have been accepted by the administrator.
  - **Requests:** displays pending requested tenants by the user accesses which have to be approved by the administrator.
  - **VBSes:** Virtual Base Stations, which interact with the controller and the created tenants, are shown in this tab. By introducing, the name and the MAC address of the corresponding eNB (srsLTE eNB in this project), VBS will be depicted in this tab.
  - **UEs:** connected users to the network detected by the controller are demonstrated in this tab with VBS, IMSI, PLMN Id and an RNTI value corresponding to each user.
  - **IMSI < = > MAC:** this tab illustrates the possibility that the IMSI of each SIM card can be mapped to the MAC address of the mobile terminal that uses this SIM Card.
  - **Users:** this tab shows the existing access types, root as administrator type and foo or bar as regular user type.

  Figure 52, summarizes the 5G-EmPOWER web interface tabs in administrator access corresponding to the LTE network:

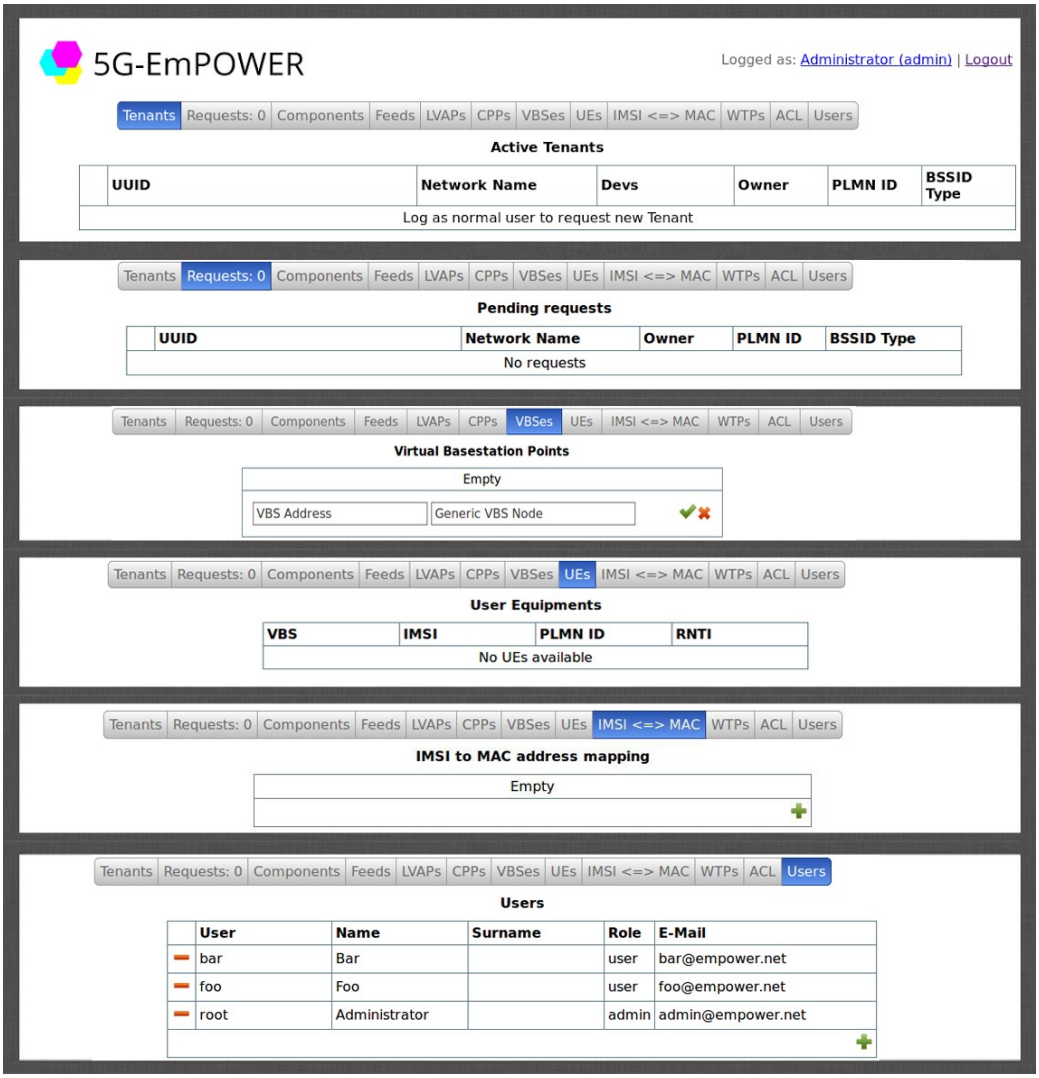


Figure 52. 5G-EmPOWER web interface tabs in administrator access.

The rest of the tabs are related to the Wi-Fi technology; thus they will not be discussed in this project (not used).

- In the case of regular user access (with username/password: foo or bar):
    - **Tenants:** the active tenants are presented in this tab. Moreover, creating tenants is performed in this tab (+ sign) with the required information (Request new Tenant in the popup window). These requested tenants will be shown as pending requested tenants in the **Requests** tab of the administrator access.
    - **Requests:** This tab displays the tenants that have been requested by the user accesses (foo or bar) and are waiting to be accepted by the administrator to become active tenants.

  Figure 53, depicts the user access tabs of the 5G-EmPOWER web interface (foo which is the same as bar):

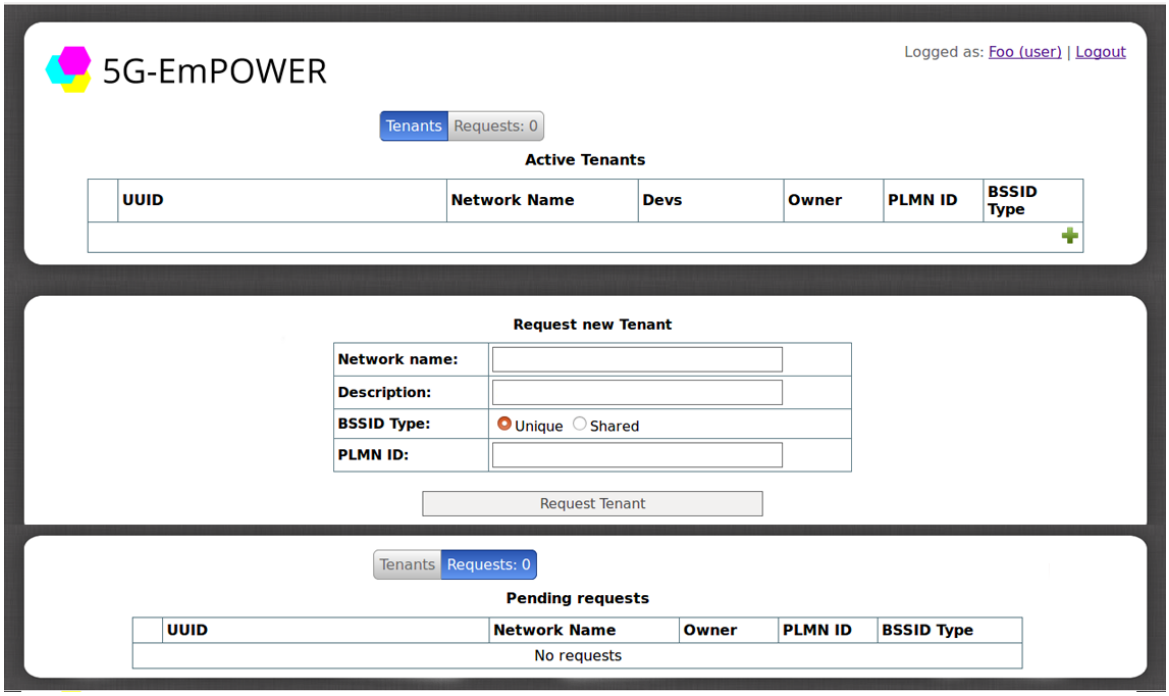


Figure 53. 5G-EmPOWER web interface tabs in user access.

4.6.3. EmPOWER Agent entity configuration

So far, OAI EPC and srsLTE eNB platforms have been correctly installed, compiled and configured to act as the core and E-UTRAN parts of the network in this project. Moreover, the 5G-EmPOWER controller has been installed and by using its web interface, users (either as administrator or regular user) have access to it. As explained before, in order to allow communication between the 5G-EmPOWER controller and the srsLTE eNB, an entity is needed known as the EmPOWER Agent. This entity, interacts with the 5G-EmPOWER controller based on a TCP protocol, known as the empower-enb-proto. Moreover, it interacts with the srsLTE eNB through the agent wrapper (known as agent) that is included (added) in the empower-srsLTE software. Figure 54 depicts the above mentioned architecture.

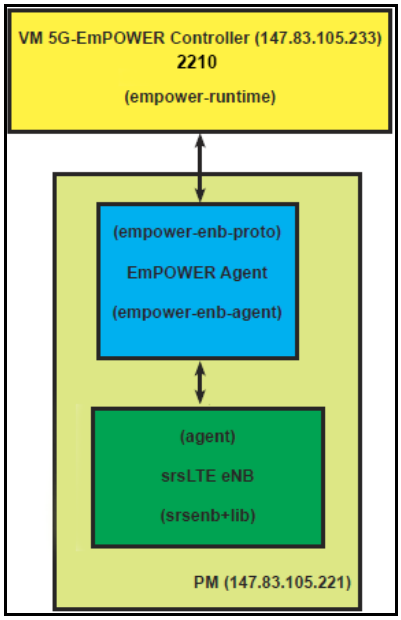


Figure 54. 5G-EmPOWER and srsLTE eNB interaction with corresponding software codes.

4.6.4. agent configuration

As mentioned previously, a wrapper (agent) is needed in order to translate the EmPOWER Agent messages into the LTE stack. This agent has been included in the open-source code provided by FuN-CreateNet, although in order to activate it there is the need to enable some specific flags during the installation process. In particular, there are two flags in two files (CMakeLists.txt and CmakeCache.txt) that enable the agent usage in the network (see Figure 55). After enabling these flags, the empower-srsLTE code has to be compiled (for detailed information refer to Annex [2]).

File: CMakeLists.txt

```
########################################################################
# Options
########################################################################

option(ENABLE_EMPOWER_AGENT  "Enable the use of EmPOWER Agent"     ON)

if(ENABLE_EMPOWER_AGENT)
  message(STATUS "Building srsENB with EmPOWER Agent.")
  add_definitions(-DHAVE_EMPOWER_AGENT)

  find_package(Emagent)
  find_package(Emproto)
endif(ENABLE_EMPOWER_AGENT)
```

File: CMakeCache.txt

```
########################################################################
# Options
########################################################################

// Enable the use of EmPOWER Agent
ENABLE_EMPOWER_AGENT : BOOL=ON

// Build srsEPC applicatiom
ENABLE_SRSEPC : BOOL= OFF
```

Figure 55. CmakeLists.txt and CMakeCache.txt files contain the flag for agent activation.

As the last step, the 5G-EmPOWER controller IP address (in our case: 147.83.105.233) and its port (2210) have to be introduced in the agent configuration file (agent.conf) as shown in Figure 56. Notice that the IP address and the port number are the same as the ones introduced in the srsLTE eNB configuration file (enb.conf).

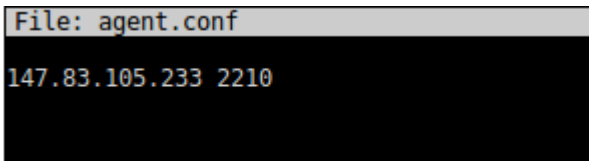


Figure 56. agent.conf file.

4.6.5. Integration of srsLTE eNB (VBS) with 5G-EmPOWER

The final step for the integration of the 5G-EmPOWER controller with the srsLTE eNB is to introduce the srsLTE eNB to the controller using the web interface. This is carried out through the introduction of a special parameter in order to allow the controller to recognize the eNB. This parameter is the eNB_id previously defined in the configuration file of the srsLTE eNB. By reviewing this file (see Figure 37), the eNB_id=0x19B can be extracted. Then via accessing the controller web interface as the administrator and introducing eNB_id=0x19B with a MAC address format (00:00:00:00:01:9B) in the VBSes tab, the integration process is accomplished. Figure 57, illustrates this process:

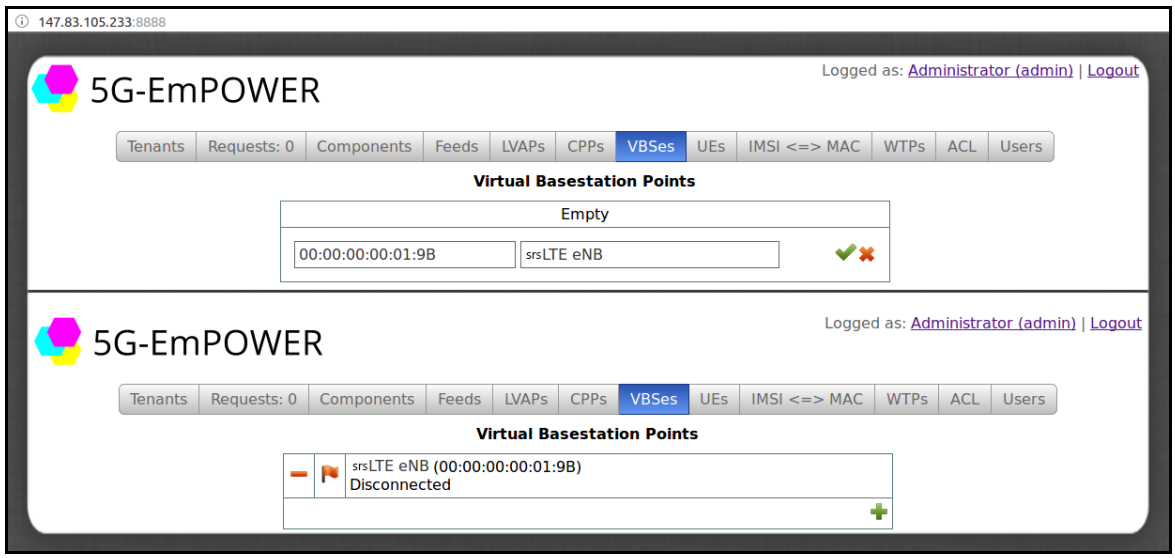


Figure 57. Creating VBS in the controller web interface.

Notice that, there is a flag symbol close to the srsLTE eNB in the web interface, which acts as an indicator of the state of the VBS. When the flag is red, it means that the network is not running or that the controller cannot identify and connect to this VBS. While the network is running, if the red flag turns to green indicates that the VBS has been identified by the controller.

4.7. Launching the network and performing connectivity tests between 5G-EmPOWER Controller and srsLTE eNB

Since all the SDN-NFV platform entities have been installed and configured, performing some connectivity tests between the 5G-EmPOWER Controller and srsLTE eNB is necessary in order to confirm that the entire system has been integrated properly. To do these tests:

- First the OAI EPC entities have to be launch (in the proper order: HSS, MME and then SPGW).
- Then, the srsLTE eNB has to be executed. With this execution the EmPOWER Agent starts the search for an existing controller.
- Finally, the 5G-EmPOWER controller has to be launched.

(See Annex [2] for the detailed information about how to run all the entities).

According to the 5G-EmPOWER platform specifications, several special messages should appear in the Linux terminal of the 5G-EmPOWER controller to verify the connection establishment between the controller and the eNB. These messages are:

1. Incoming connection message from the host where the srsLTE eNB has been implemented on.
2. Receiving Hello messages from the srsLTE eNB (actually generated by the EmPOWER Agent) and changing the VBS mode (disconnected => connected).
3. Message in which the controller sends a request to the eNB (actually to the EmPOWER Agent) to know its capabilities.
4. Message in which the controller receives a response from the eNB (actually generated by the EmPOWER Agent) with its capabilities and changing the VBS mode (connected => online).
5. Message in which the controller sends a request to the eNB (actually to the EmPOWER Agent) in order to ask for the report of the possible UEs connected to the eNB.

Figure 58, displays these messages based on the corresponding number that each message has been explained above.

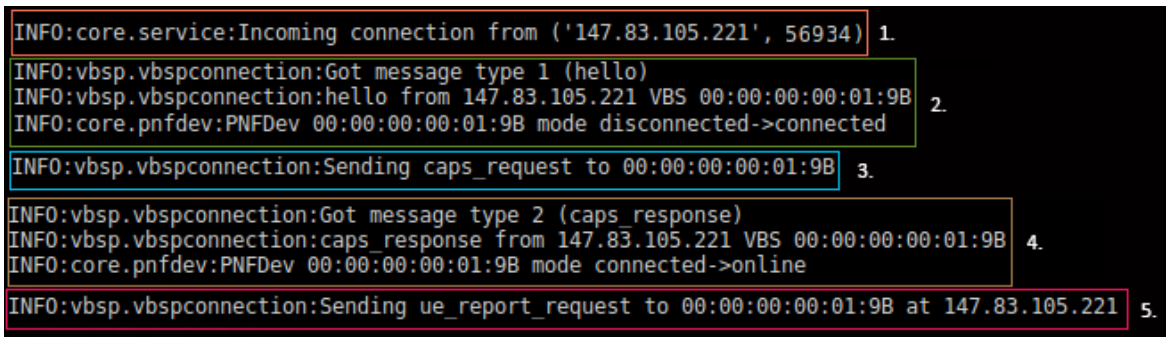


Figure 58. Linux terminal of 5G-EmPOWER.

Obtaining these messages confirms that the connection establishment between the srsLTE eNB and the 5G-EmPOWER controller has been carried out correctly.

In the VBSes tab of administrator access, the created VBS now has a green flag indicating that the controller has detected the VBS (eNB) while running (see Figure 59):

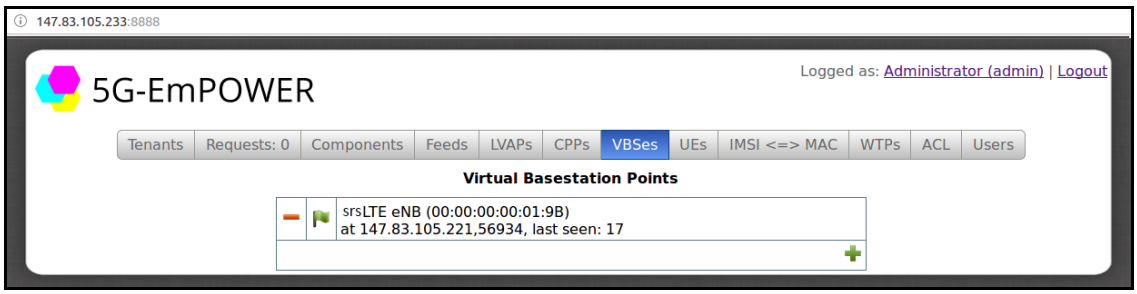


Figure 59. VBSes tab of administrator access while srsLTE eNB is running.

4.8. Implementation of a tenant slice and performing connectivity tests

Creating a slice or tenant is the last procedure in order to have a complete functional system.

The different steps of the procedure used for the creation of a tenant with PLMN Id=21491, is explained in the following:

1. By using one of the regular user accounts (foo or bar), in the **Tenants** tab of it and by clicking on the green color +, the **Request new Tenant** popup window will be accessed (see Figure 60).

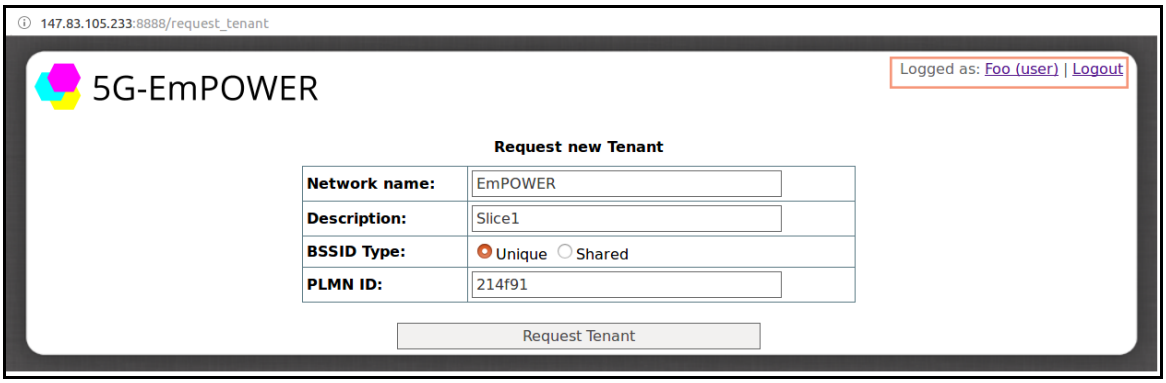


Figure 60. Creating a new tenant.

Then in this window the main parameters of the tenant have to be determined:

- A name and an optional description have to be assigned to the tenant to be created.
- Basic Service Set Identifier (BSSID) is a unique identifier for a particular BSS within an area and it is included in all the packets of the BSS to identify them as part of that network. So it has to be set as **Unique**.
- Define the PLMN Id of the network, which results from the combination of the MCC and the MNC. It is essential to point out that MNC in this project has been defined with 2 digits since the MNC in Spain has only 2 digits.

However, the PLMN Id can obtain up to 3 digits. The PLMN Id format of the network which integrates with EmPOWER controller has to be defined based on the MNC with 3 digits. So it is necessary to fill (pad) the MSB (Most Significant Bit) of the MNC with 1111 which corresponds to 0xf in hexadecimal format. Therefore, the final format of the PLMN Id will be: **PLMN Id=MCCfMNC** and consequently in this project: **PLMN Id=214f91**.

2. Pending requested tenant created by the regular user appears in user`s Requests tab and is waiting to be approved by the administrator in its Requests tab (see Figure 61):

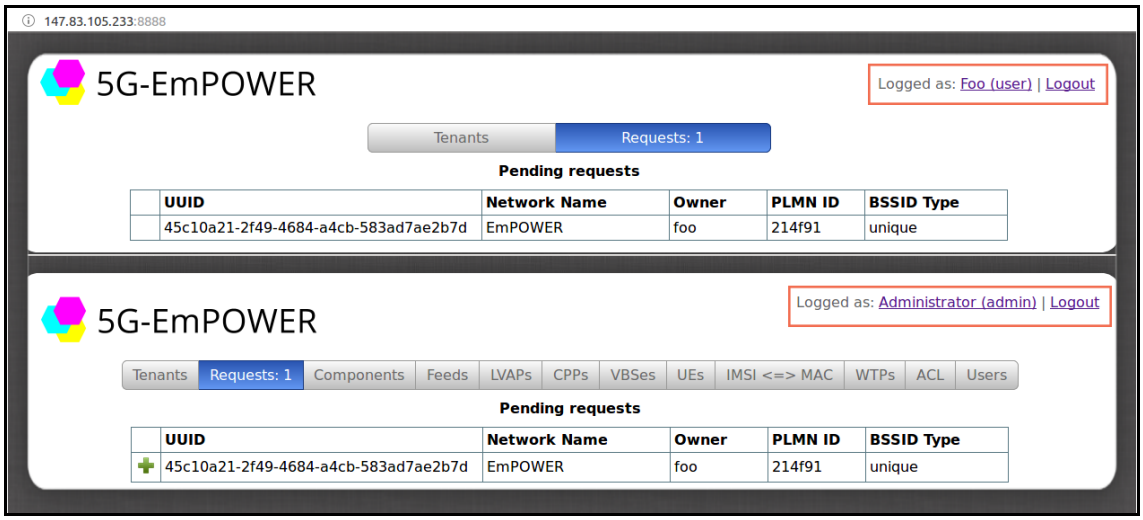


Figure 61. Requested tenant in Requests tab of user and Administrator accounts.

3. Once the tenant is accepted by the Administrator, it becomes an active tenant (see Figure 62):

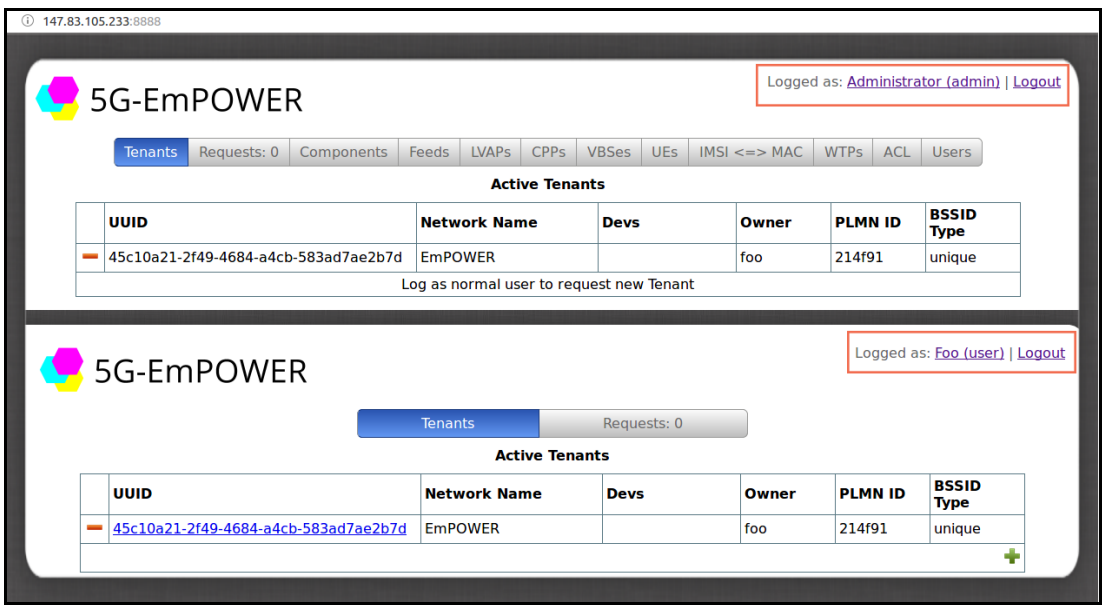


Figure 62. Displaying an active tenant in the Tenants tab of the Administrator and user accounts.

4. Finally, by clicking on the **UUID** section of **Tenants** tab in the user account, a new window will open where in the **VBSes** tab, the corresponding VBS (srsLTE eNB defined previously) has to be associated with this tenant (see Figure 63):

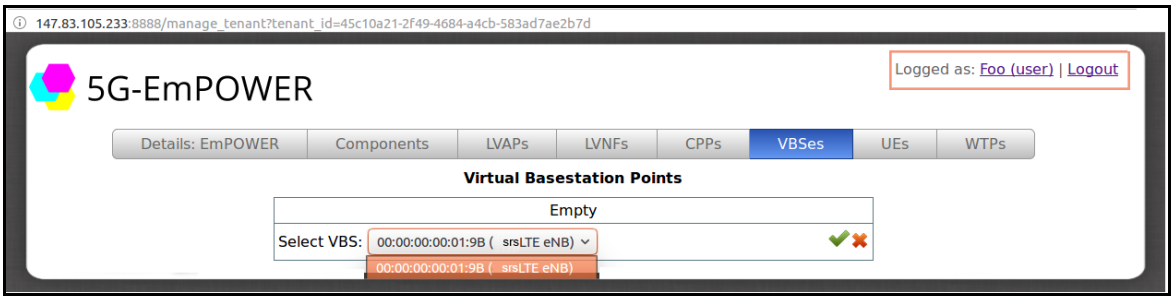


Figure 63. VBS association with the created tenant.

In this way, the tenant will be created and associated with the VBS (eNB).

4.9. Launching the network and perform UE connectivity tests

As the final test, the UE connectivity with the created tenant has to be tested. As such, all the network entities (OAI EPC, srsLTE eNB and 5G-EmPOWER controller) have to be launched (in the same way explained before) and in this phase, the UEs (with previously defined APN) have to be able to connect to the network and specifically to the created tenant with PLMN Id=21491. It has to be mentioned that since the network is able to handle and assign radio resources at up to 16 UEs simultaneously connected to it, the multi-connectivity tests are worth to be performed to confirm also this aspect of the network.

In that respect, the messages appearing in the corresponding Linux terminals and also the 5G-EmPOWER web interface while performing multi-connectivity tests for UEs connected to the created tenant, are illustrated in the following Figures:

- **In Linux terminals**
    - HSS terminal: both SIM Card IMSIs have been authenticated and are ready to connect to the network. (cell phone+SIM Card with IMSI=214910000009911, dongle+SIM Card with IMSI=214910000009915 -see Figure 64):

Figure 64. HSS terminal messages in the multi-connectivity test.

- MME terminal: the terminal shows two connected UEs with the assigned radio resources to them (see Figure 65):

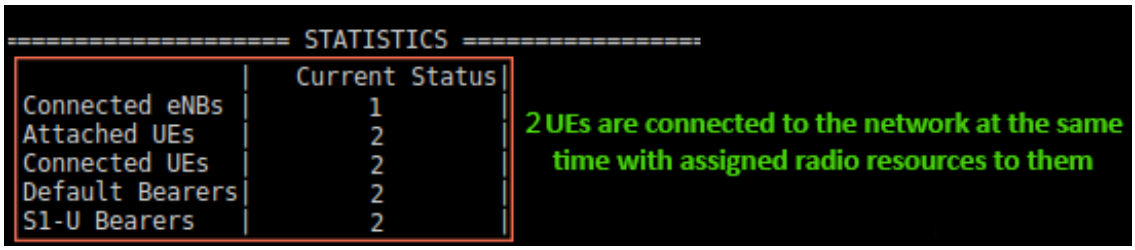


Figure 65. MME terminal statistics table in the multi-connectivity test.

- srsLTE eNB terminal: this terminal displays the rnti values and also physical parameters corresponding to the UEs (see Figure 66):

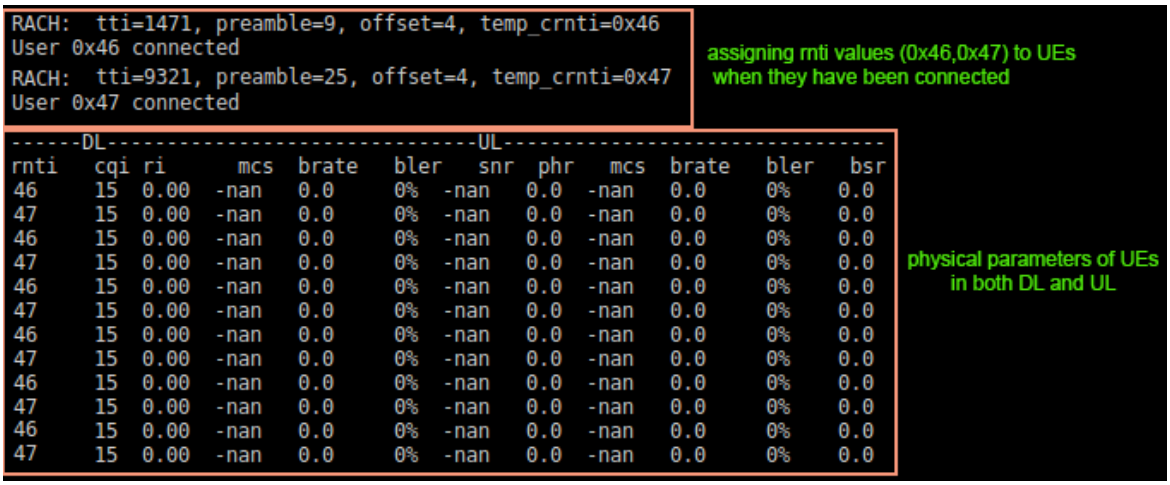


Figure 66. srsLTE eNB terminal in the multi-connectivity test.

- 5G-EmPOWER terminal: this terminal after displaying the initial messages discussed before, first show the created tenant, which is now activated, and then display the information about the two UEs connected to the network (see Figure 67):

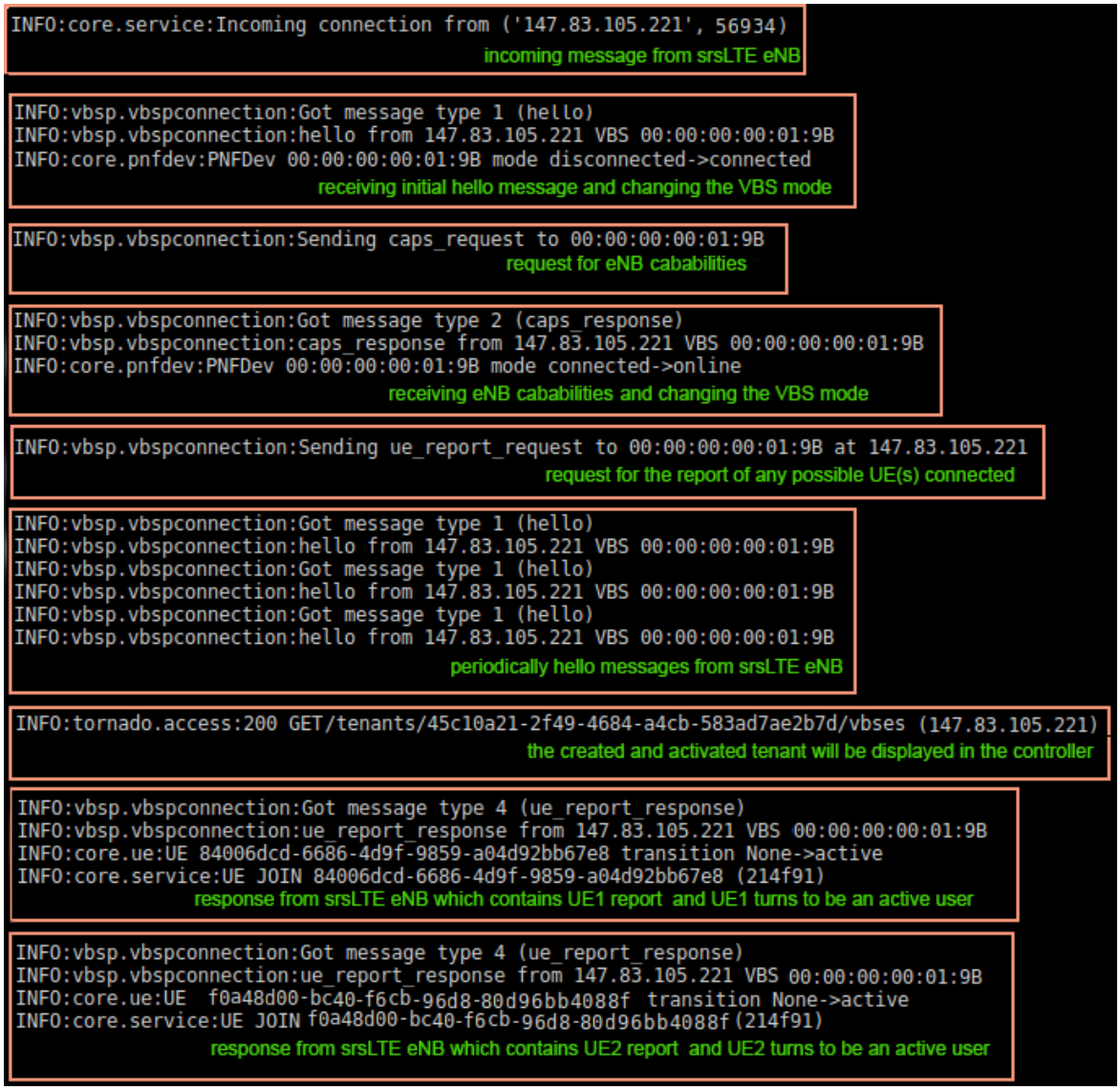


Figure 67. 5G-EmPOWER controller terminal in the multi-connectivity.

- **In 5G-EmPOWR web interface**

  The information of the active tenant (Network Name: EmPOWER) and the UEs connected to that tenant (rnti values and PLMN Id) appear in the administrator and regular user accounts (see Figure 68):

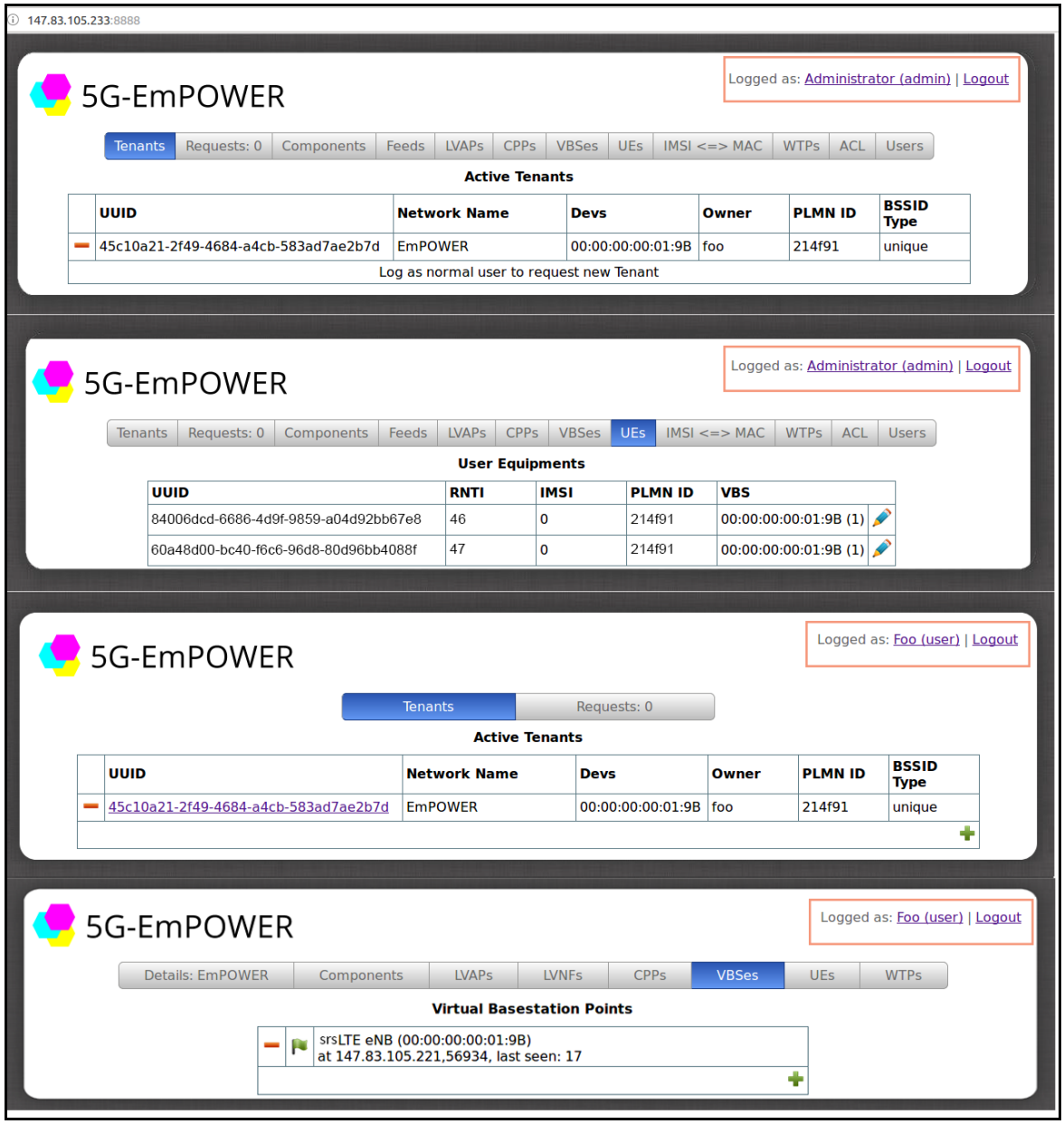


Figure 68. 5G-EmPOWER controller web interface in the multi-connectivity test.

At the end of this section it is worth to mention that:

- Even if the IMSI field is present in the report of any possible UEs (ue_report_response) message, the particular EmPOWER Agent implementation can decide to leave it filled with zero, since specific operations are necessary to bind IMSI to RNTI, and they are not part of the standard yet [26]. This is why the corresponding section in the **UEs** tab of the administrator account has been filled with zero.
- In the case of multi-connectivity, no negative effects of degrading the quality of services offered by the network to the 2 connected UEs have been observed.

4.10. Network operation in debugging mode

The operation of the network in debugging mode is discussed in this section. Since quite common errors might happen during the integration of the different entities (such as EPC and 5G-Empower controller) with the srsLTE eNB, it is highly recommended to configure, compile and run the system in such a way that allows us to back trace the possible errors in order to fix them.

Even though this section covers only the execution part, detailed information of the whole process including the configuration and compilation processes can be found in Annex [3].

Network operation in debugging mode generates more messages than in the normal mode. These messages, in the case of empower-srsLTE (agent section) and EmPOWER Agent (empower-enb-agent, empower-enb-proto) are presented in different ways. The following diagram explains these ways (see Figure 69):

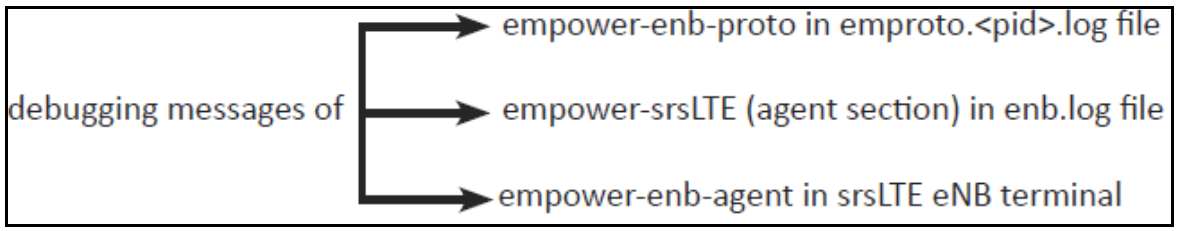


Figure 69. Ways of printing debugging messages.

In the following, each case is presented with an example:

- **emower-enb-proto**

  Different .c files in empower-enb-proto software print their messages inside the emproto.<pid>.log file by using the **ep_dbg_dump** function. <pid> is the process identifier and changes with every execution of the network so, the previous log files will not be replaced by any new ones. An example of the use of the **ep_dbg_dump** function in the ephello.c file is illustrated in Figure 70. Let us notice that this file forms part of the hello exchanging message process with the controller.

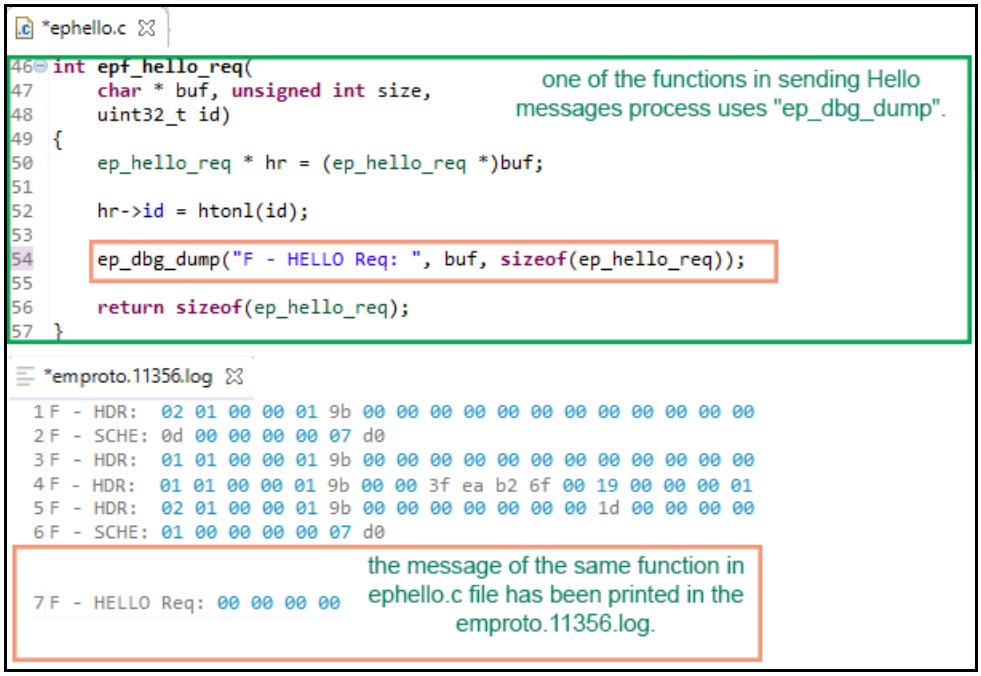


Figure 70. ep_dbg_dump function in ephello.c and the printed message.

- **emower-srsLTE (agent/wrapper section)**

  The empower_agent.cc is the file corresponding to the agent (wrapper) that is included in the empower-srsLTE. Let us remind that the agent is responsible for the exchange and translation of messages between the srsLTE eNB and the EmPOWER Agent (empower-enb-agent section). There is the possibility to print agent messages inside the enb.log file by using the **Debug** function (see Figure 71).

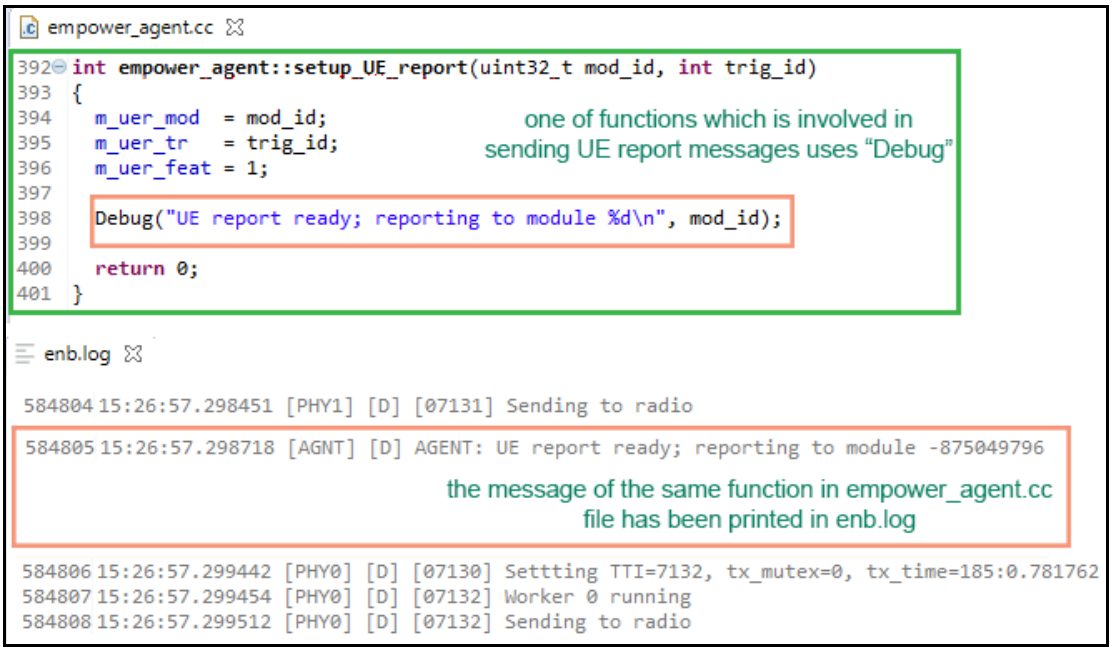


Figure 71. Debug function in epower_agent.cc and the printed message.

- **empower-enb-agent**

  Different .c files in the empower-enb-agent software can print debugging messages in the srsLTE eNB terminal by using one of the **EMDBG** or **EMLOG** functions. net.c belongs to these family of .c files and the use of **EMDBG** function is illustrated in Figure 72:

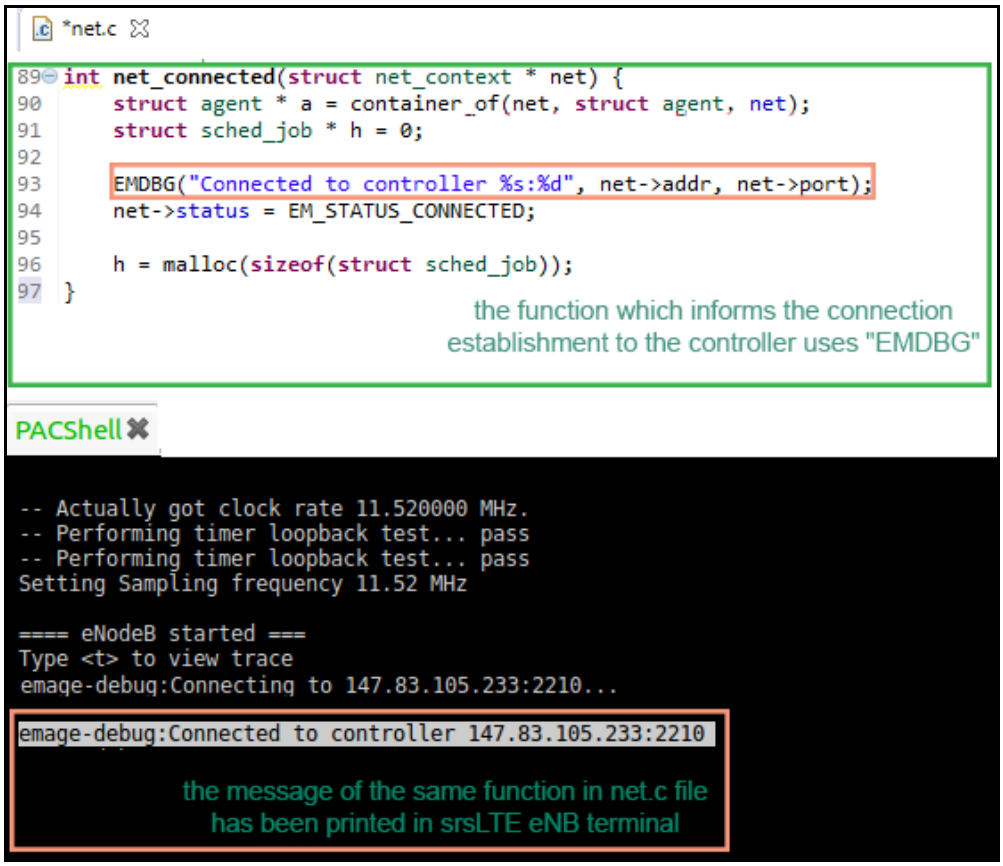


Figure 72. EMDBG function in net.c and the printed message.

As such, in order to be able to perform the debugging process it is necessary to enable the network to operate in debugging mode (during the compilation process) and use the above described functions. The following Figure is a useful example to understand the process of sending hello messages to the Controller when the network operates in debugging mode (see Figure 73).

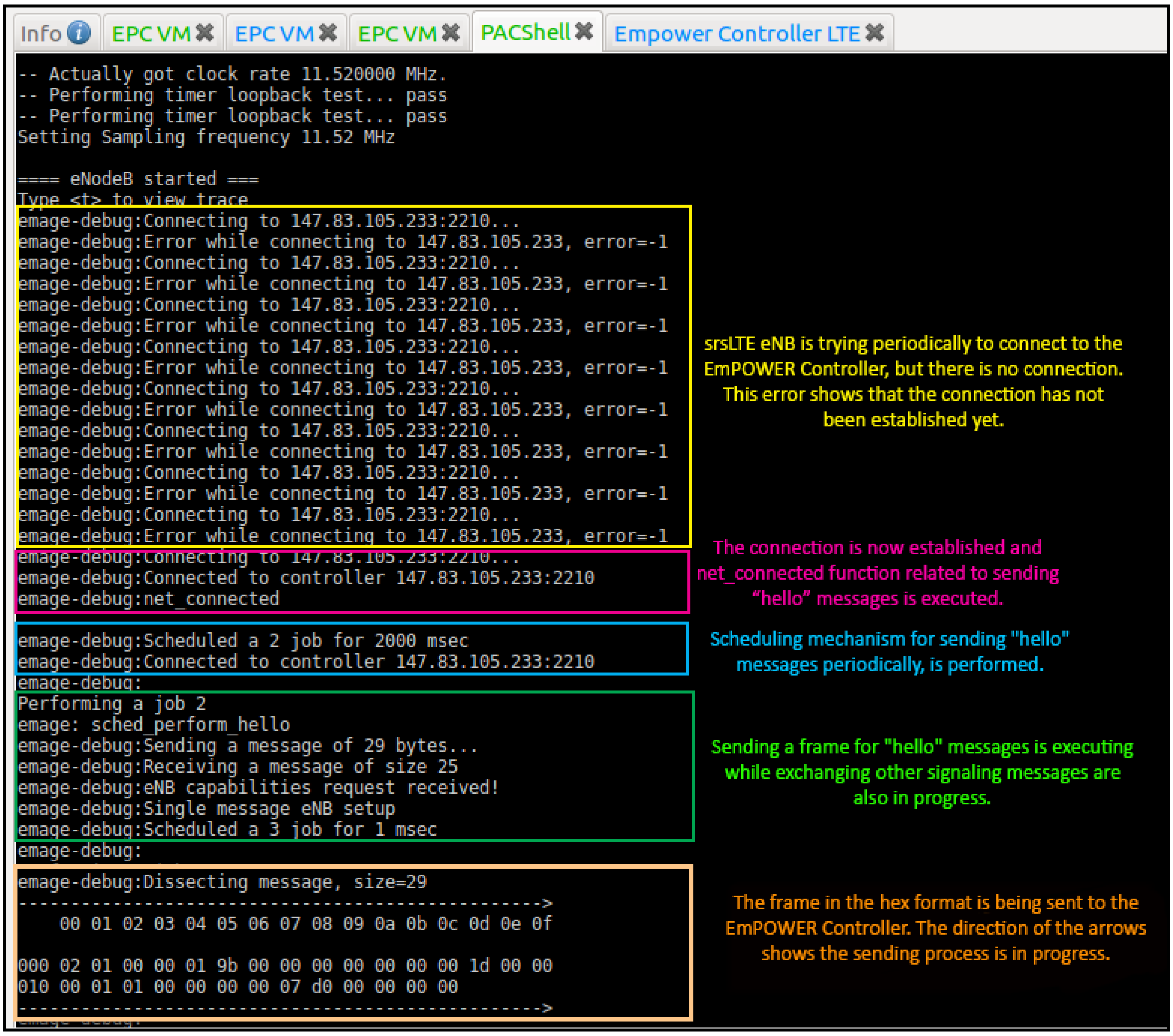


Figure 73. Detailed messages while sending hello messages to the controller.

4.11. Code block diagram and code modifications

Since the installation, compilation and configuration processes have been covered in detailed and the debugging process has been explained in order to enable the network to identify and resolve possible errors, the next part of the project is dedicated to the code analysis. The final objective is to be able to perform modifications on code, compile them and perform the debugging processes whenever it is necessary. An overview of the different partitions of the code, as well some modifications in the code are presented in this chapter.

4.11.1. Code block diagram

Based on the code classification that has been done by the SRS LTE developers [27], srsenb and agent code, written in C/C++, are defining a **superclass** called **srsenb**. Then according to the LTE protocol stack, srsenb superclass has been split out into **five parts** which are: core, three main layers, and the agent. The contents of these parts are known

as subclasses of the srsenb superclass. The following diagram displays these five parts with their corresponding subclasses (see Figure 74):

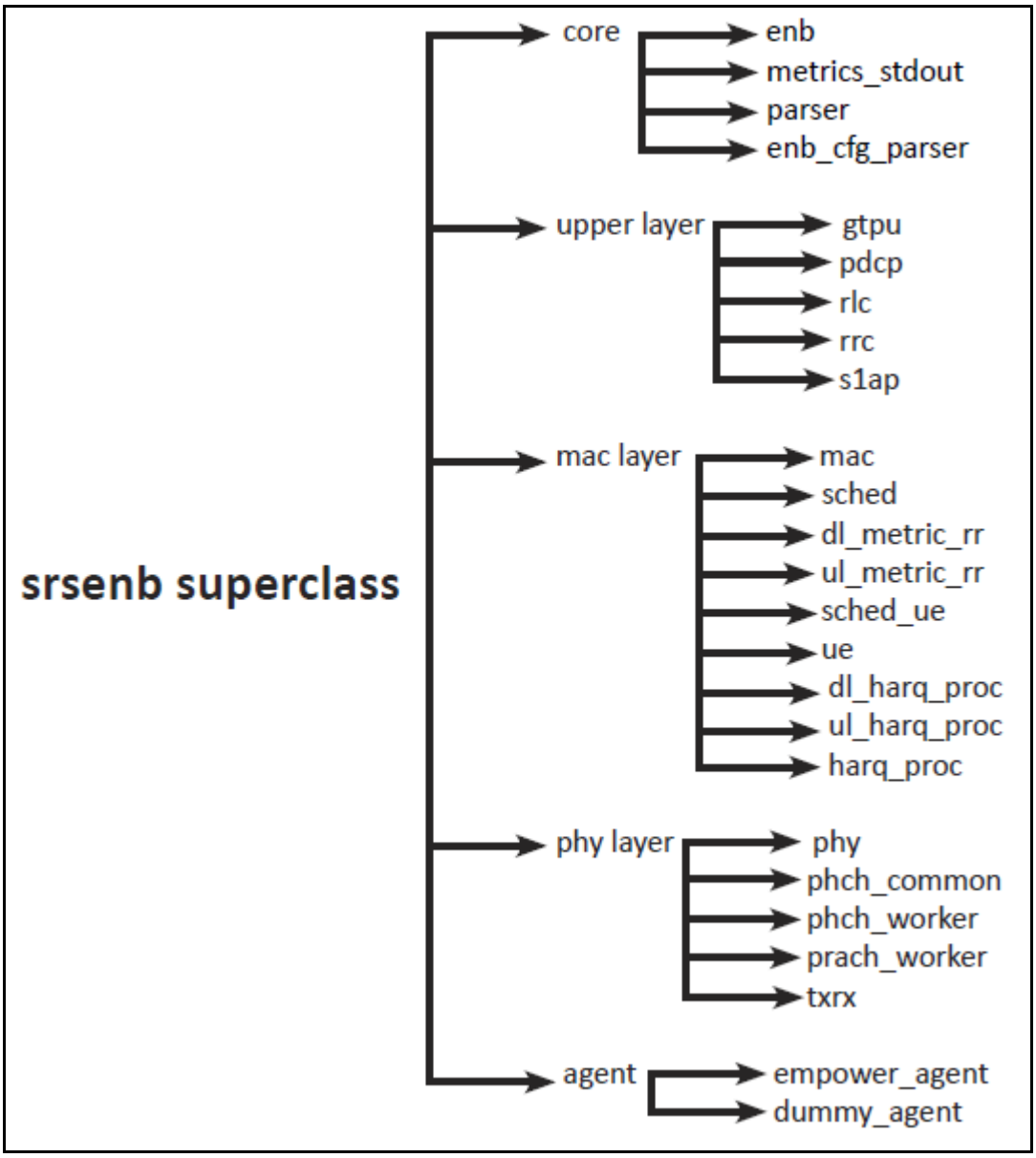


Figure 74. srsenb superclass.

Please refer to Annex [4], where the detailed information about these subclasses, their attributes and functions can be found.

Another aspect that has to be taken into account is how these subclasses can interact with each other and what the relationships between them are. A block diagram has been created (see Figure 75) demonstrating these relationships. Let us notice that the diagram is a simplified version of a quite complex one, since these subclasses, in turn, have their own subclasses too.

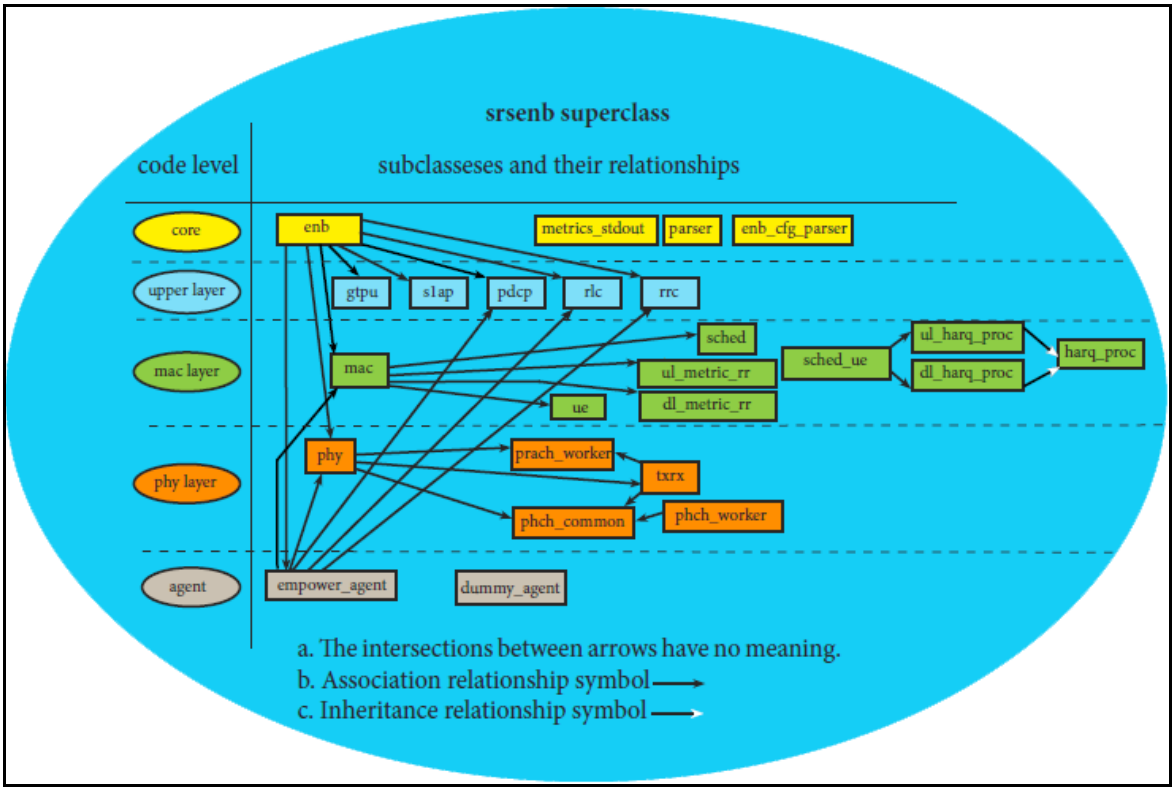


Figure 75. srsenb superclass block diagram.

4.11.2. Code modifications

A brief explanation of the specifications for exchanging information between the srsLTE eNB and the 5G-EmPOWER controller through the EmPOWER Agent is presented in this section. Moreover, the process for the addition of a "keep connection alive" message, is described. This message is similar to the *hello* message, although in this case it is called "*hola*" message.

By studying the documentations provided by create-net developers [28] [29], analyzing and classifying them, the following results have been obtained:

- The **EmPOWER Agent** is the one who initiates the communication with the controller by sending periodically *hello* messages which are coming from srsLTE eNB.
- All kind of messages have a standard frame structure (defined by the empower protocol), depicted in Figure 76:
-

Figure 76. Frame structure of EmPOWER protocol.

- **Master header:** All messages contain a Master header. It includes features such as: message type (in the case of *hello*, it has been defined as a scheduled event so:0x02, it will be discussed in the following part), protocol version, eNB_id (0x19B extracted from enb.conf file), cell_id (0x01 extracted from enb.conf file), module_id (created to be used in the controller side for issuing the query), message length, and sequence number (used by the srsLTE eNB and the controller for each module_id).
- **Event header field:** It identifies the family of actions that can occur in the system. The event header field includes: type of event (single:0x01, scheduled:0x02, and trigger:0x03), direction of the message is being sent (the one which sends request:0x00, the one which sends reply:0x01), operation type (successful:0x01, fail:0x02), interval (value in msec), and padding. There are three kind of events [28]:
    - **single event**: simple and standalone events requested by the EmPOWER controller and notified back immediately by the EmPOWER Agent or vice-versa, such as *eNB capability* messages.
    - **scheduled event**: events performed multiple times (every specific time interval). These events allow the EmPOWER controller/Agent to interchange periodical messages, such as *hello* messages.
    - **trigger event**: events which should ideally enable/disable a functionality in the agent. They work with a threshold or particular actions that can happen in a not predictable way, such as *UE report* messages.
- **Action field:** They are independent from events, and only describe what is exactly necessary to be performed.
- **Data field:** The actual data information which is to be sent in the request or response messages is placed in this section (in case of *hello* message, no data has to be sent).

In order to implement the *hola* message, new functions have been introduced, as well as functions already implemented in the EmPOWER Agent have been used (or slightly modified). The reason for this is that there are some functions, such as the ones responsible for the creation of the frame to be transmitted (based on the EmPOWER protocol), that only need to be called or modified slightly in order to perform the necessary actions for the *hola* message transmission. In addition, in order to implement the *hola* message, new code modules had to be added in the original code. Finally, the whole software was recompiled and verified that the incorporation of the *hola* message was successfully carried out through connection tests and debugging processes.

Figures 77 and 78 briefly explain the procedures followed in the project in order to implement the hola message with the corresponding defined/modified functions (see Annex [5], where all the corresponding functions have been included). A brief description is given in the following:

- *core.c, sched.c* and *net.c* files form part of the EmPOWER Agent and are located in the *empower-enb-agent*. *ephola.c* is a module that has to be included in the EmPOWER protocol definition, thus it is introduced in the *empower-enb-proto*. Let us notice that *net.c* logically contains the necessary routines for handling the communication with the controller. In particular, *net.c* is responsible for the message reception/transmission, identification of the different types of messages

(e.g. triggered), call of the necessary functions in order to handle a received message and in cases that a response has to be sent in the controller, it calls the necessary functions in order to send the response. It is worth of noting that the *net.c* also handles and monitors the connection with the controller. The file *sched.c* is responsible for scheduling the various procedures (or jobs) in the CPU.

- According to the EmPOWER protocol, each message (and more particular each action) is distinguished from both the controller and agent by the "packet type identifier". As such, in the *epdefs.c* and *eptypes.c* files, it has to be defined: EP_ACT_HOLA=13. Let us notice that the number 13 has been chosen since this is the 13th packet type has been created.
- In order to be able to distinguish the *hola* type of "job" its definition must be introduced in the *epdefs.c* and *eptypes.c* files (part of the EmPOWER protocol). As such, we introduce JOB_TYPE_HOLA = 9, since this job is the 9th to be defined. In this way, the agent will know what it has to perform (what functions to call) in order to send the message. After doing all the modifications, the whole system needs to be recompiled.
- After sending the message, the same process will be repeated, since the scheduling of the network processes is in a loop mode.

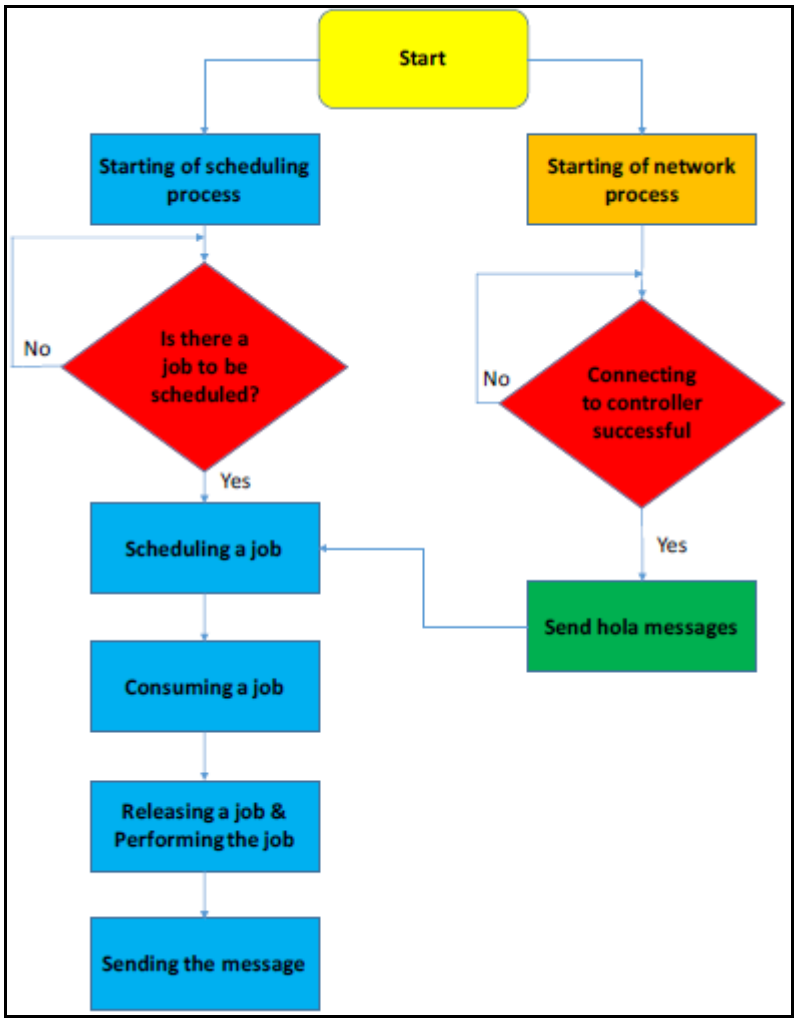


Figure 77. Sending *hola* message flow diagram (Agent side).

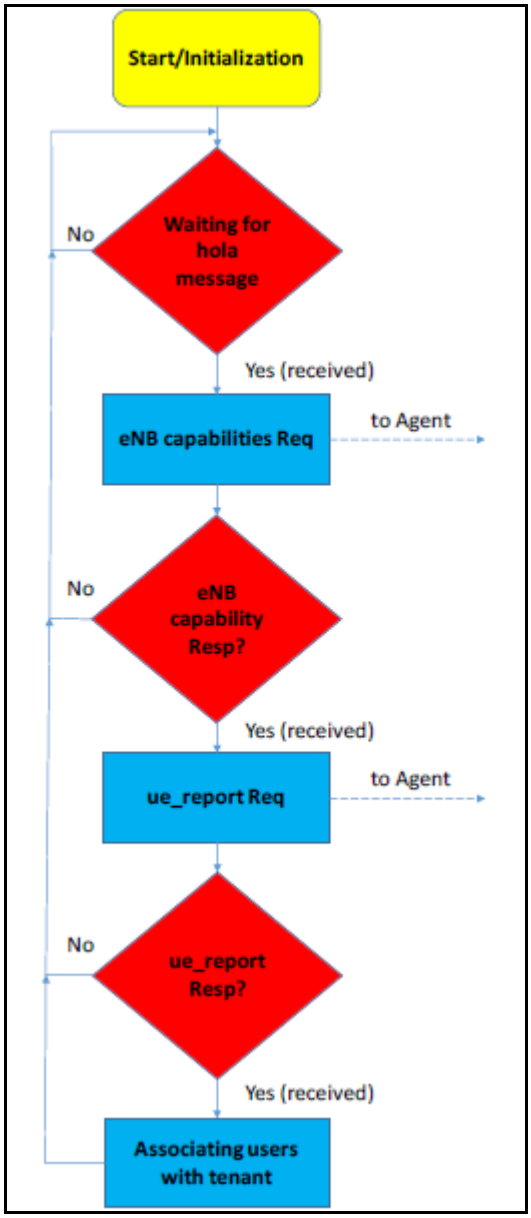


Figure 78. Handing the incoming *hola* message flow diagram (controller side).

The following Figures illustrate the results of performing all these procedures in both srsLTE eNB (Figure 79) and 5G-EmPOWER controller (Figure 80) terminals:

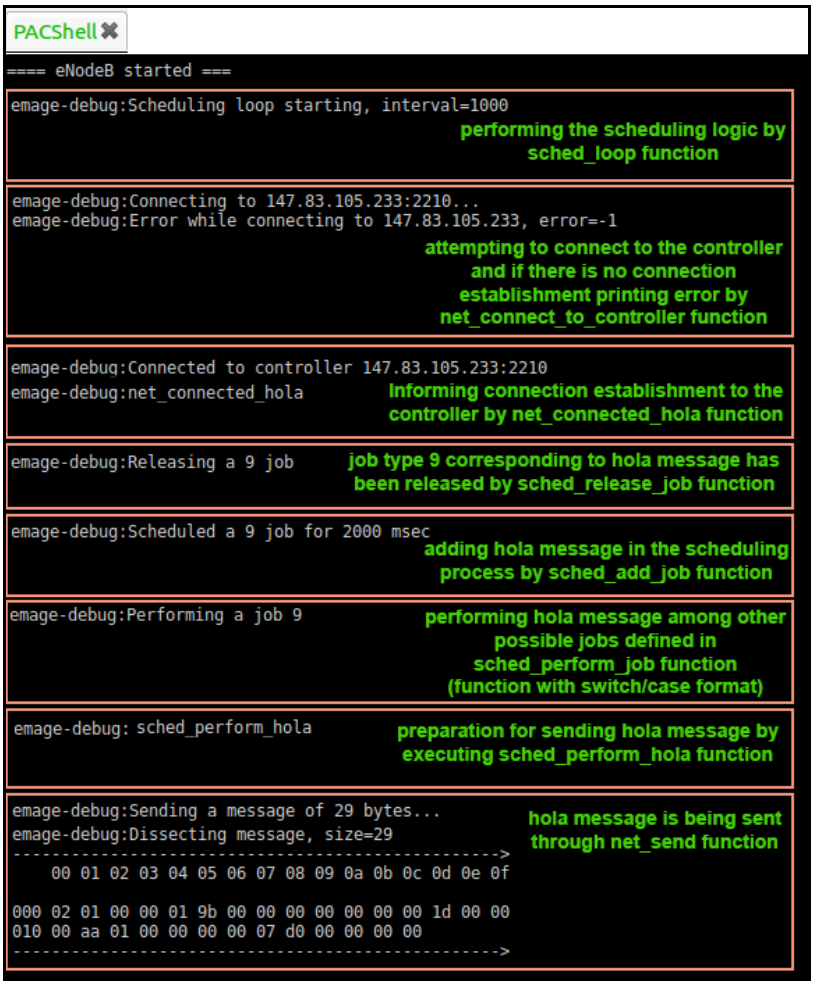


Figure 79. Process of sending *hola* message in srsLTE eNB terminal.

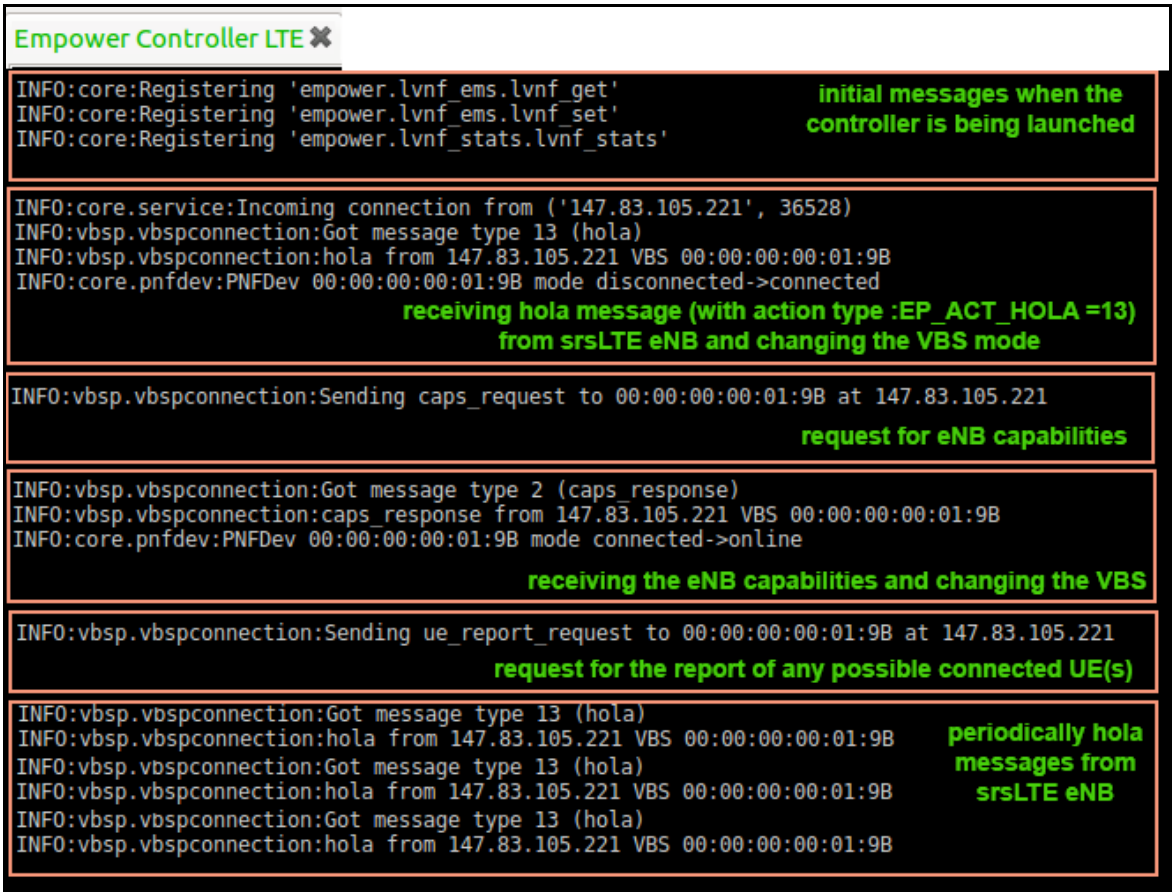


Figure 80. Process of receiving *hola* message in 5G-EmPOWER controller terminal.

5. Conclusions and future development

As noted the flow of data information is growing dramatically and the pattern of processing information in traditional networks will not be able to satisfy the user demand. Therefore, research activities in the data communication area are focused on possible solutions that can cope with the above mentioned issues by keeping in mind that such new solutions have to:

- be compatible with the traditional networks.
- propose a new way of managing data information inside the networks that makes the network management and data processing easier.
- be capable to be implemented without forcing too many expenses to the network operators.
- be flexible to interacted with various involved platforms.
- improve user experience compared with currently existing technologies by offering services with significant quality levels.
- possess a high level of security and reliability.
- provide opportunities for innovating solutions related to the data processing and management.

These benefits have been summed up in SDN-NFV technology.

Integrating SDN-NFV technology with cellular networks such as LTE, brings all the above mentioned advantages to the mobile communication area by applying orchestration of intelligent services and dynamic resource control and management.

SDN-NFV technologies have been involved in this project in the form of three main platforms: OpenAirInterface as the core, Software Radio Systems as the E-UTRAN, and 5G-EmPOWER as the controller. Moreover, the whole system has been studied in order to provide the basis for the future implementation of RAN Slicing approaches.

In this way, during the project, all the platforms have been integrated and several tests carried out in order to verify the connectivity between the various elements. The next step included the creation of a tenant in order to test that the controller can associate users with a specific slice of resources. In fact, by using the controller web interface, creating and configuring tenants have been accomplished in a simple and intuitive way. Various connectivity tests with different kinds of UEs (cell phones and dongles) have been performed, including multi-user scenarios. In all cases, the radio resources with appropriate quality have been provided for the UEs.

Network operation in debugging mode has been covered in detail, since it allows displaying messages with additional information in order to validate the correct operation of the network when code modifications are performed or to find solutions when problems occur.

Finally, a new message that keeps alive the connection between the controller and the eNB (through the EmPOWER Agent) has been implemented and validated through debugging mode and connectivity tests.

The future developments for networks similar to the network discussed in this project, could be mainly focused on the contribution to the implementation of the proposed

framework in [30]. This framework establishes the new blocks of information, configuration descriptors, and extended protocol features to be introduced within E-UTRAN for the realization of RAN slicing. In this way enabling multiple RAN slices with potentially different radio protocol behaviours, as well as different levels of guaranteed network resources and isolation to be concurrently multiplexed over the same cell, would be possible. Besides this significant target which requires much attempt and time, there would be several possibilities for developing the project such as:

- Implementation of a weight management algorithm in the controller in order to maximize the utilization of VBSes resources.
- Creating new or customized messages to be exchanged between srsLTE eNB and the EmPOWER controller such as a customized user measurement message.
- Taking the advantage of srsLTE EPC to play the role of core entity in the network and compare the performance of srsLTE EPC with OAI EPC.
- For next releases that include more features and capabilities, more complex scenarios can be performed like: handling more than one eNB or EPC.

Bibliography

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020. USA; 2018:1-6. [Online] Available:

https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf].

[2] Sallent, O., Perez-Romero, J., Ferrus, R. and Agusti, R. (2017). On Radio Access Network Slicing from a Radio Resource Management Perspective. IEEE Wireless Communications, 24(5), pp.166-174.

[3] Nunes B, Mendonca M, Nguyen X, Obraczka K, Turletti T. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. IEEE Communications Surveys & Tutorials. 2014;16(3):1617-1634. doi:10.1109/surv.2014.012214.00180.

[4] Hewlett Packard, Asia-Pacific. (June 2014). Virtualisation of Network Functions and the SDN. Improving the Economics of the Network.: AJTDE - Vol 2, No 2 - [Online] Available:

https://telsoc.org/ajtde/2014-06-v2-n2/a41.

[5] Margaret Chiosi. (October 22-24, 2012). Network Functions Virtualisation:An Introduction, Benefits, Enablers, Challenges & Call for Action. - [Online] Available:

http://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[6] Network Functions Virtualisation:Overview by datacomm [Online] Available:

https://www.datacomm.co.id/en/telco/nfv/.

[7] M.J. Martin. (June 1, 2017). Dynamic Networks: SDN and NFV [Online] Available:

https://vividcomm.com/2017/06/01/dynamic-networks-sdn-and-nfv/.

[8] Prayson Pate. (March 30, 2013). NFV and SDN: What's the Difference? [Online] Available:

https://www.sdxcentral.com/articles/contributed/nfv-and-sdn-whats-the-difference/2013/03/.

[9] Tinku Rasheed. (02 December 2015). Cellular Software Defined Network – a Framework. [Online] Available:

https://www.researchgate.net/publication/273380717_Cellular_Software_Defined_Network_-_a_Framework.

[10] Agustí Comes R, Bernardo F, Casadevall F, Ferrús R, Pérez-Romero J, Sallent O. LTE: NUEVAS TENDENCIAS EN COMUNICACIONES MÓVILES. [Madrid]: Fundación Vodafone España; 2010.

[11] Paul Sutton. Software Radio Systems. The Open-Source SDR LTE Platform for First Responders. [Online] Available:

https://www.gnuradio.org/wp-content/uploads/2017/12/Paul-Sutton-SRS-OpenFirst.pdf.

[12] Alcatel-Lucent. The LTE Network Architecture. A comprehensive tutorial. [Online] Available:

http://www.cse.unt.edu/~rdantu/FALL_2013_WIRELESS_NETWORKS/LTE_Alcatel_White_Paper.pdf.

[13] Advantages of LTE | Disadvantages of LTE | Long Term Evolution. Rfwireless-worldcom. 2018. [Online] Available:

http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-LTE.html.

[14] GitLab. (2018). oai / openairinterface5G. [online] Available: https://gitlab.eurecom.fr/oai/openairinterface5g

[15] OPENAIRINTERFACE/openair-cn. GitHub. 2018. [online] Available: https://github.com/OPENAIRINTERFACE/openair-cn.

[16] A Little LTE for You & Me: Build Your Own LTE Network on a Budget. [online] Available:

https://medium.com/@CableLabs/a-little-lte-for-you-me-build-your-own-lte-network-on-a-budget-86e9d38ed575.

[17] USRP B200/B210. Santa Clara, CA; 2018:1-2. [online] Available: https://www.ettus.com/content/files/b200-b210_spec_sheet.pdf.

[18] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton,Pablo Serrano, Cristina Cano, Douglas J. Leith. (15 February 2016). srsLTE: An Open-Source Platform for LTE Evolution and Experimentation. [online] Available:

https://arxiv.org/pdf/1602.04629.pdf.

[19] Software Radio Systems. [online] Available:

http://www.softwareradiosystems.com/.

[20] Riggio R. 5G-EmPOWER: Multi-access Edge Computing Operating System. Empowercreate-netorg. 2018. [online] Available:

http://empower.create-net.org/.

[21] Roberto Riggio-CREATE-NET. (16 October 2016). Demo: The EmPOWER Mobile Network Operating System. [online] Available:

http://www.robertoriggio.net/papers/wintech2016_demo.pdf.

[22] GitLab. (2018). 5g-empower. [online] Available:

https://github.com/5g-empower/empower-enb-proto/.

[23] sysmocom site - sysmoUSIM-SJS1 SIM + USIM Card (10-pack). [online] Shop.sysmocom.de. Available:

http://shop.sysmocom.de/products/sysmousim-sjs1.

[24] Mcc-mnc.com. (2018). Most up to date list of MCC and MNC codes: mobile country codes – mobile network codes. [online] Available:

http://www.mcc-mnc.com/.

[25] LTE: Tracking Area (TA) and Tracking Area Update (TAU). [online] Available:

https://www.netmanias.com/en/post/blog/5930/lte-tau/lte-tracking-area-ta-and-tracking-area-update-tau.

[26] GitLab. (2018). 5g-empower. [online] Available:

https://github.com/5g-empower/empower-enb-proto/blob/master/documentation/v1/uerep.txt.

[27] GitLab. (2018). 5g-empower. [online] Available:

https://github.com/5g-empower/empower-srsLTE/tree/agent/srsenb.

[28] GitLab. (2018). 5g-empower. [online] Available:

https://github.com/5g-empower/empower-enb-proto/tree/master/documentation/v1/protocols.txt.

[29] GitLab. (2018). 5g-empower. [online] Available:

https://github.com/5g-empower/empower-enb-agent/tree/master/documentation.

[30] Sallent, O., Perez-Romero, J., Ferrus, R. and Agusti, R. (2018). On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration. IEEE Communications Magazine. [online] Available:

https://www.researchgate.net/publication/322376498_On_5G_Radio_Access_Network_Slicing_Radio_Interface_Protocol_Features_and_Configuration

Annex

This Annex contains the links to access all the users` manuals which have been provided during the project progress.

[1] Config-Run user manual for srsLTE eNB with OAI EPC:

https://drive.google.com/file/d/1Wtb0tugi-b3WQ50us1EJO6I9zLIzagZ_/view?usp=sharing

[2] Config-Run user manual of EmPOWER Controller for integration with srsLTE eNB:

https://drive.google.com/file/d/1_SAWNizv9eRxud10a4F7hKMnnBkupZDT/view?usp=sharing

[3] Config-Run user manual for srsLTE eNB in debugging mode:

https://drive.google.com/file/d/1gNvLhE2TBoCOWoPEeOIcxtezfVj6q71b/view?usp=sharing

[4] Code Diagram tutorial for srsLTE eNB and EmPOWER Agent:

https://drive.google.com/file/d/1rEduFFA3AEyzfT5DzsKbvpG8__95xmVL/view?usp=sharing

[5] Modified-Created functions for code modification (creating hola message):

https://drive.google.com/file/d/1zwfNbU_MSK-Pgav2KliQ1_F0zW4ZsFSc/view?usp=sharing

Glossary

APN	Access Point Name
CAPEX	Capital Expenditures
CC	Country Code
CN	Core Network
CPP	Click Packet Processor
C-RAN	Cloud RAN
CSDN	Cellular SDN
eNB	evolved NodeB
EPC	Evolved Packet Core
E-UTRAN	Evolved Terrestrial Radio Access Network
FDD	Frequency Division Duplexing
FuN	Future Networks Unit
GRCM	Mobile Communications Research Group
GSM	Global System for Mobile Communications
GUMMEI	Globally Unique MME Identity
HSS	Home Subscriber Server
ICCID	Integrated Circuit Card Identifier
IMSI	International Mobile Subscriber Identity
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
LVAP	Light Virtual Access Point
LVNF	Light Virtual Network Function
MCC	Mobile Country Code
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity

MNC	Mobile Network Code
MSP	Mobile Service Provider
MT	Mobile Terminal
MVNO	Mobile Virtual Network Operator
NFV	Network Function Virtualization
OAI	OpenAirInterface
OFDMA	Orthogonal Frequency Division Multiple Access
OPc	Operator key or code
OPEX	Operating Expenditures
OSA	OpenAirInterface Software Alliance
P-GW	PDN Gateway
PLMN Id	Public Land Mobile Network Identifier
QoS	Quality of Service
RAN	Radio Access Network
SC-FDMA	Single Carrier Frequency Division Multiple Access
SDN	Software Defined Networking
S-GW	Serving Gateway
TA	Tracking Area
TAI	List Tracking Area Identity List
TCP	Transmission Control Protocol
TDD	Time Division Duplexing
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
USRP	Universal Software Radio Peripheral
VBS	Virtual Base Station
VM	Virtual Machine
VNF	Virtualized Network Function