TRABAJO FIN DE GRADO

INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

# Prototipo de una estación base 4G usando *Open Air Interface*


**Autor**

Francisco García Espigares

**Directores**

Jorge Navarro Ortiz

José Carlos Segura Luna




Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación


—

Granada, Septiembre de 2017


# Prototipo de una estación base 4G usando *Open Air Interface*

**Realizado por**
Francisco García Espigares

**Directores**
Jorge Navarro Ortiz
José Carlos Segura Luna

**Departamento**
Teoría de la Señal, Telemática y Comunicaciones

Granada, Septiembre de 2017

# Prototipo de una estación base 4G usando Open Air Interface

Francisco García Espigares

Palabras clave

LTE (*Long Term Evolution*), OAI (*Open Air Interface*), USRP (*Universal Software Defined Radio*), 4G, Sysmocom, VMware, OAISIM (*Open Air Interface System Emulation*), USIM (*Universal Subscriber Identity Module*)

Resumen

En los últimos años y de forma más acelerada, el mundo de las TIC (*Tecnología de la Información y la Comunicación*) está llevando a cabo una continua transformación a través de múltiples aspectos como la aparición de nuevas tecnologías o la implantación de innovaciones en los sistemas *software* y *hardware*, llevando consigo oportunidades que posibilitan el nacimiento de nuevos mercados. Dicha transformación tiene lugar por las exigencias de la sociedad.

En el año 2016, el informe anual de la UIT (Unión Internacional de Telecomunicaciones) estimaba que alrededor de 7.000 millones de personas viven en una zona cubierta por una red móvil celular 2G, una cifra que casi equivale a la población mundial. Además, con la aparición de la cuarta generación de redes móviles durante los últimos tres años, éstas redes de banda ancha proporcionan cobertura a casi 4.000 millones de personas. El informe pronosticó que, para finales de 2016, el número de suscriptores pasaría a 3.600 millones, representando el 53 % de la población mundial y mejorando así la conectividad a Internet.

Por estos motivos, el presente proyecto pretende ilustrar la arquitectura de la red móvil de cuarta generación, asentando los conocimientos sobre los equipos que la componen y su interconexión. El objetivo principal de este proyecto consiste en la puesta en marcha de una celda LTE (*Long Term Evolution*) utilizando para ello el *software* libre Open Air Interface, un dispositivo USRP (*Universal Software Radio Peripherical*), y un PC comercial. Esto conlleva la correcta instalación y configuración de la red de acceso radio o E-UTRAN (*Evolved Universal Terrestrial Radio Access Network*), además de la red troncal o EPC (*Evolved Packet Core*). Dicho proyecto concluirá con la evaluación de dicha plataforma, tanto desde el punto de vista de las señales radio como de la señalización empleada.

# Prototype of a 4G base station using Open Air Interface

Francisco García Espigares

Keywords

LTE (*Long Term Evolution*), OAI (*Open Air Interface*), USRP (*Universal Software Defined Radio*), 4G, Sysmocom, VMware, OAISIM (*Open Air Interface System Emulation*), USIM (*Universal Subscriber Identity Module*)

Abstract

In recent years and more rapidly, the ICT world (*Information and Communication Technology*) is undergoing a continuous transformation through multiple aspects such as the emergence of new technologies or the implementation of innovations in *software* and *hardware* systems, bringing with it opportunities that allow the emergence of new markets. This transformation takes place by the demands of society.

In 2016, the annual ITU (*International Telecommunication Union*) report estimated that about 7 billion people live in an area covered by a 2G mobile cellular network, a number almost equal to the world population. In addition, with the emergence of the fourth generation of mobile networks over the last three years, these broadband networks provide coverage to almost 4 billion people. The report predicted that by the end of 2016 the number of subscribers would rise to 3.6 billion, representing 53 % of the world's population and thus improving Internet connectivity.

For these reasons, the present project intends to illustrate the architecture of the fourth generation mobile network, establishing the knowledge about its entities and their interconnection. The main objective of this project is to set up an LTE cell (*Long Term Evolution*) using the free software Open Air Interface, a USRP (*Universal Software Radio Peripherical*), and a commercial PC. This entails the correct installation and configuration of the E-UTRAN (*Evolved Universal Terrestrial Radio Access Network*) network, in addition to the EPC (*Evolved Packet Core*). This project will conclude with the evaluation of this platform, both from the point of view of the radio signals and the signaling procedures.

Yo, **Francisco García Espigares**, alumno de la titulación GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI XXX, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo. Francisco García Espigares

Granada a Septiembre de 2017.

D. **Jorge Navarro Ortiz**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

D. **José Carlos Segura Luna**, Profesor del Área de Teoría de la Señal y Comunicaciones del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado:

***Prototipo de una estación base 4G usando Open Air Interface***

ha sido realizado bajo su supervisión por **Francisco García Espigares**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a Septiembre de 2017.

**Los directores:**

Fdo. Jorge Navarro Ortiz

Fdo. José Carlos Segura Luna

# Agradecimientos

Me gustaría aprovechar esta oportunidad, que es única en la vida; ya que todo lo que un día decides comenzar con coraje y determinación, un día termina, pero con una sensación y un sentimiento completamente distinto. En primer lugar, no tengo palabras para describir lo orgulloso que estoy de mis padres Francisco y Carmen, por la educación que me han transmitido a mi y a mis hermanos. Ellos han conseguido de mi el hombre que soy. También, quiero agradecerle este mérito a mis hermanos Rafa y Eduvigis, siempre estaré para lo que necesitéis, a mis abuelos, a mis tíos, a mis primos, y a mi familia, ya que con su cariño y apoyo han logrado que hoy esté escribiendo estas palabras. Estoy más tranquilo porque puedo asegurar que se ha cumplido un sueño, y es el comienzo de otro; este final no tendría el mismo sabor sin el apoyo que he recibido en la última etapa, ni sin la ilusión que me aporta mi compañera de viaje Ana, te quiero.

A mis amigos, tanto dentro de la ETSIIT como los de la infancia, especialmente a José Ortega, por el apoyo recibido en los primero años de carrera y por ser como un hermano; a Manu, por tener que soportarme durante toda la carrera hasta llegar al punto de la convivencia, y llegar a ser tan importante como un miembro de mi familia.

Además, me gustaría agradecerles el enorme apoyo que he recibido de mis tutores Jorge y José Carlos, por darme la oportunidad de realizar este proyecto con cara al futuro profesional. Siempre dispuestos a preocuparse por mis dudas; agradecerles sus consejos, su enseñanza, y por darme esa confianza.

Gracias a todos.

# Índice general

# Lista de Figuras

# Lista de Tablas

# Acrónimos

**2G** *Segunda Generación Móvil.*

**3G** *Tercera Generación Móvil.*

**3GPP** *Third Generation Partnership Project.*

**ACK** *Acknowledgement.*

**ACLR** *Adjacnt Channel Leakage Ratio.*

**AM** *Acknowledged Mode.*

**APN** *Access Point Name.*

**AS** *Access Stratum.*

**AuC** *Authentication Centre.*

**AV** *Authentication Vector.*

**BCCH** *Broadcast Control Channel.*

**BCH** *Broadcast Channel.*

**BSC** *Base Station Controller.*

**BW** *Bandwidth.*

**CCCH** *Common Control Channel.*

**CN** *Core Network.*

**CNMC** *Comisión Nacional de los Mercados y la Competencia.*

**CRF** *Charging Rules Function.*

**DCCH** *Dedicated Control Channel.*

**DL** *Down Link.*

**DL-SCH** *Dowlink Shared Channel.*

**DMRS** *Demodulation Reference Signal.*

**DRB** *Data Radio Bearer.*

**DTCH** *Dedicated Traffic Channel.*

**EIR** *Equipment Identity Register*.

**eNB** *evolved Node Base*.

**EPC** *Evolved Packet Core*.

**E-UTRAN** *Evolved UMTS Terrestrial Radio Access Network*.

**EVM** *Error Vector Magnitude*.

**FDD** *Frequency Division Duplexing*.

**FFT** *Fast Fourier Transform*.

**FQDN** *Fully Qualified Domain Name*.

**GPRS** *General Packet Ratio Service*.

**GSM** *Global System for Mobile Communication*.

**GTP-U** *GPRS Tunnelling Protocol - User Plane*.

**GUMMEI** *Globally Unique Mobile Management Entity Identifer*.

**GUTI** *Globally Unique Temporary Identity*.

**HARQ** *Hybrid Automatic Repeat Request*.

**HLR** *Home Location Register*.

**HPLMN** *Home Public Land Mobile Network*.

**HSS** *Home Subscriber Server*.

**IMEI** *International Mobile Equipment Identity*.

**IMS** *IP Multimedia Subsystem*.

**IMSI** *International Mobile Subscriber Identity*.

**IP** *Internet Protocol*.

**LTE** *Long Term Evolution*.

**MAC** *Medium Access Control*.

**MBMS** *Multimedia Broadcast and Multicast System*.

**MCC** *Mobile Country Code*.

**MCCH** *Multicast Control Channel*.

**MCH** *Multicast Channel*.

**MIB** *Master Information Block*.

**MME** *Mobility Management Entity*.

**MNC** *Mobile Network Code*.

**MTCH** *Multicast Traffic Channel.*

**NACK** *Negative Acknowledgement.*

**NAS** *Non-Access Stratum.*

**NAT** *Network Address Translation.*

**OAI** *Open Air Interface.*

**OAISIM** *Open Air Interface System Emulation.*

**OFDM** *Orthogonal Frequency Division Multiple.*

**OFDMA** *Orthogonal Frequency Division Multiple Access.*

**OSA** *Open Air Interface Software Alliance.*

**PBCH** *Physical Broadcast Channel.*

**PCCH** *Paging Control Channel.*

**PCFICH** *Physical Control Format Indicator Channel.*

**PCRF** *Policy and Charging Rules Funcion.*

**PCH** *Paging Channel.*

**PDCCH** *Physical Downlink Control Channel.*

**PDCP** *Packet Data Convergence Protocol.*

**PDF** *Policy Decision Function.*

**PDN-GW** *Packet Data Network Gateway.*

**PDP** *Packet Data Protocol.*

**PDSCH** *Physical Downlink Shared Channel.*

**PDU** *Packet Data Unit.*

**P-GW** *Packet Data Network Gateway.*

**PHICH** *Physical Hybrid ARQ Indicator Channel.*

**PLMN** *Public Land Mobile Network.*

**PMCH** *Physical Multicast Channel.*

**PRACH** *Physical Random Access Channel.*

**PSD** *Power Spectral Density.*

**PSS** *Primary Synchronization Signal.*

**P-TMSI** *Packet Temporary Mobile Subscriber Identity.*

**PUCCH** *Physical Uplink Control Channel.*

**PUSCH** *Physical Uplink Shared Channel.*

**QAM** *Quadrature Amplitude Modulation.*

**QoS** *Quality of Service.*

**QPSK** *Quadrature Phase Shift Keying.*

**RACH** *Random Access Channel.*

**RAN** *Radio Access Network.*

**RAT** *Radio Access Technology.*

**RB** *Resource Block.*

**RE** *Resource Element.*

**RLC** *Radio Link Control.*

**RNC** *Radio Network Controller.*

**RNL** *Radio Network Layer.*

**RRC** *Radio Resource Control.*

**S1-AP** *S1 Application Protocol.*

**S1-MME** *S1 Mobility Management Entity.*

**SAP** *Service Access Points.*

**SC-FDMA** *Single Carrier - Frequency Division Multiple Access.*

**SCTP** *Stream Control Transmission Protocol.*

**SDF** *Service Data Flows.*

**SDR** *Software Defined Radio.*

**SFN** *System Frame Number.*

**SGSN** *Serving GPRS Support Node.*

**S-GW** *Serving Gateway.*

**SI** *System Information.*

**SIB** *System Information Blocks.*

**SRB** *Signaling Radio Bearers.*

**SRS** *Sounding Reference Signal.*

**SSS** *Secondary Synchronization Signal.*

**TA** *Tracking Area.*

**TAI** *Tracking Area Identifier.*

**TB** *Transport Blocks*.

**TCP** *Transport Control Protocol*.

**TDD** *Time Division Duplexing*.

**TEID** *Tunnel Endpoint Identifier*.

**TIC** *Tecnología de la Información y la Comunicación*.

**TLS** *Transport Layer Securtiy*.

**TM** *Transparent Mode*.

**TNL** *Transport Network Layer*.

**UDP** *User Datagram Protocol*.

**UE** *User Equipment*.

**UL** *Up Link*.

**UL-SCH** *Uplink Shared Channel*.

**UM** *Unacknowledged Mode*.

**UMTS** *Universal Mobile Telecommunications System*.

**USIM** *Universal Subscriber Identity Module*.

**USRP** *Universal Software Radio Peripheral*.

**UTRAN** *UMTS Terrestrial Radio Access Network*.

**X2-AP** *X2 Application Protocol*.

# Capítulo 1

# Introducción

## 1.1. Contexto de las redes móviles

El sector de las tecnologías de la información y las telecomunicaciones o *Tecnología de la Información y la Comunicación* (TIC) abarca prácticamente todos los sectores de producción y consumo. Los dispositivos móviles incorporan más y mejores servicios además de mejorar la calidad de las comunicaciones. El avance de la tecnología en los últimos años ha sido tan voraz que un solo dispositivo abarca la complejidad que tan sólo unos años atrás no era posible tenerla sin disponer varios dispositivos. Además, la revolución de los servicios a través de Internet y la demanda constante de los usuarios, hace posible que infinidad de servicios estén disponibles a nuestro alcance a través de nuestro teléfono móvil; desde poder convertirse en un mando a distancia, en una cámara de fotos digital, convertirse en un sistema de navegación por satélite, e incluso permitirnos realizar transferencias bancarias. Todos estos servicios han ido acompañados de la evolución que han sufrido las redes móviles, con las distintas generaciones que han puesto a nuestra disposición nuevos avances y tecnologías.

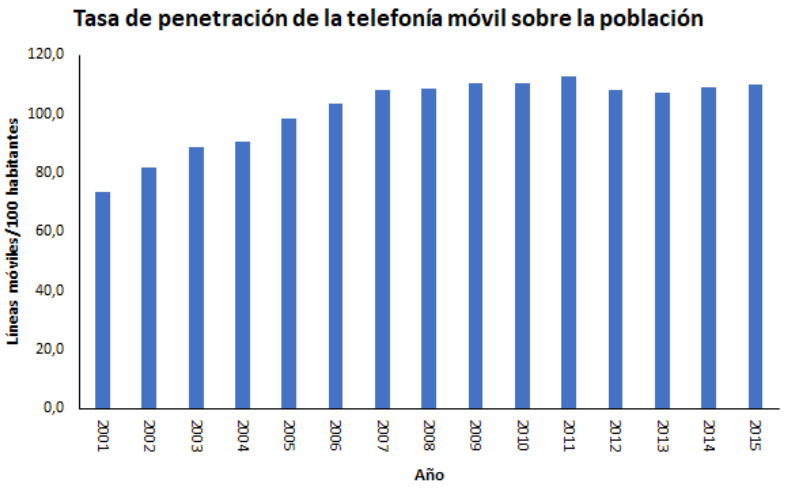



Figura 1.1: Evolución de las líneas móviles en España [7].

La sociedad está en constante transformación y los dispositivos móviles han tenido, tienen, y seguirán teniendo un papel muy importante a la hora de reflejar esa evolución. Cada día, el mercado de las TIC está cada vez más integrado en nuestra sociedad. Un claro reflejo son los datos estadísticos que nos revela la *Comisión Nacional de los Mercados y la Competencia* (CNMC). En la figura 1.1, podemos ver la evolución de

las líneas móviles cada 100 habitantes desde el año 2001 hasta el 2015; a partir del año 2006 había más líneas móviles que habitantes en España y a partir del año 2014 se aprecia un ligero ascenso en el número de lineas por habitante. Seguramente en los años venideros esta relación continué aumentando debido al nacimiento de nuevos mercados como el del *IoT* (*Internet of Things*).

Las redes móviles de cuarta generación, última generación desplegada en la mayoría de países desarrollados como es el caso de España, no han parado de expandirse desde su lanzamiento en el año 2012. La tecnología *Long Term Evolution* (LTE) fue desarrollada por la organización *Third Generation Partnership Project* (3GPP), ya que detectaron la necesidad de asegurar la competitividad de los sistemas 3G y asegurar las necesidades de los usuarios que demandaban más calidad y mayores velocidades para los diferentes servicios ofrecidos.

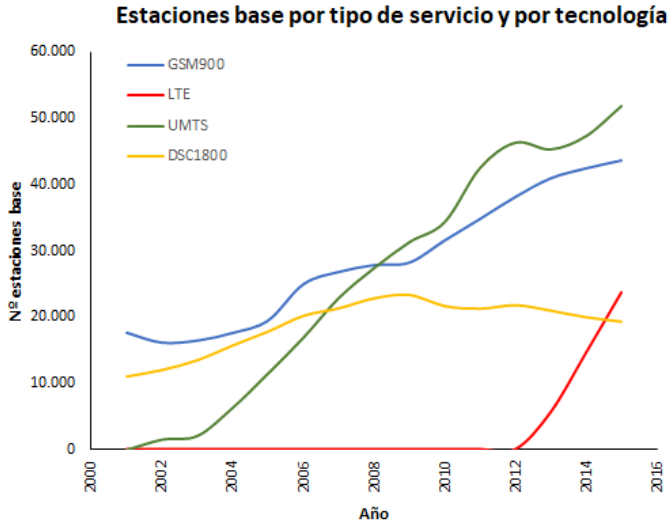



Figura 1.2: Despliegue de estaciones base [7].

En la figura 1.2 observamos, según el informe anual de la CNMC (Comisión Nacional de los Mercados y la Competencia), el número de estaciones base instaladas en España de la distintas tecnologías a las que tenemos acceso. Aquí se ve reflejado el impacto que ha tenido la tecnología LTE en España, ya que en sus primeros años ha superado en número de estaciones base desplegadas en DSC1800 (*Digital Cellular System*). El despliegue de estaciones base de *Universal Mobile Telecommunications System* (UMTS), tuvo un descenso coincidiendo con el lanzamiento de la cuarta generación de redes móviles (LTE), aunque en el año 2015 continúa en ascenso al igual que LTE. Podemos destacar el estancamiento de las redes móviles de tecnologías anteriores como GSM900, que se prevé que en los años próximos sufrirá un descenso del número de estaciones base, dando paso al nacimiento y la implantación de la quinta generación de redes móviles.

Desde la página oficial del Ministerio de Energía, Turismo y Agenda Digital se puede consultar la información de cobertura de banda ancha en España a nivel nacional, autonómico, provincial y municipal, agregada por operador y desglosada por tecnología. En la figura 1.3 tenemos el mapa de cobertura provincial para la tecnología LTE a mediados del año 2016. En este mapa podemos observar que casi la totalidad de las

provincias tienen una cobertura por encima del 75 %, lo que supone que España es uno de los países que lideran el despliegue de redes 4G.

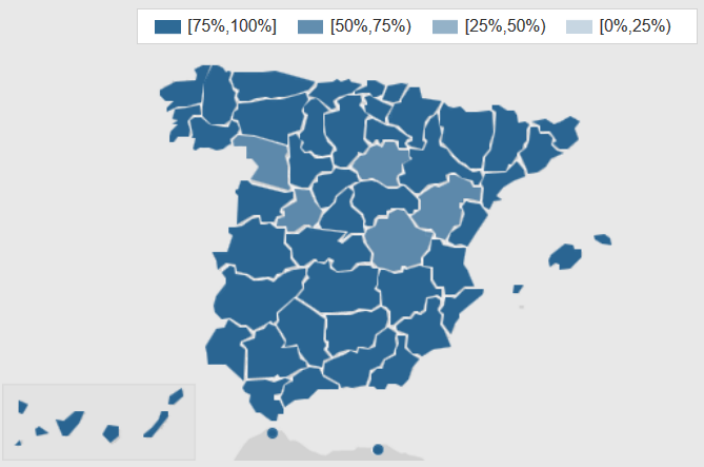



Figura 1.3: Mapa de cobertura de tecnología LTE en España,año 2016 [5].

### 1.1.1. Reparto del espectro radioeléctrico

El Cuadro Nacional de Atribución de Frecuencias (CNAF) detalla el ordenamiento del espectro, en el se indican las atribuciones a los servicios radioeléctricos y los usos de las distintas bandas de frecuencia en España. Actualmente, las frecuencias de telefonía móvil para las redes de cuarta generación en España son las que se detallan en la tabla 1.1.

| **Frecuencia (MHz)** | **Banda** | **Frecuencia UL (MHz)** | **Frecuencia DL (MHz)** | **Uso** |
|---|---|---|---|---|
| 800 | 20 | 832-862 | 791-821 | Redes móviles 4G/LTE (dividendo digital) |
| 900 | 8 | 880-915 | 925-960 | Redes móviles 3G y GSM |
| 1500 | 32 | - | 1452-1492 | Próximamente a subasta pública para redes 4G |
| 1800 | 3 | 1710-1785 | 1805-1880 | Redes móviles 4G/LTE y GSM |
| 2100 | 1 | 1920-1980 | 2110-2170 | Redes móviles 3G |
| 2600 | 7 | 2500-2570 | 2620-2690 | Redes móviles 4G/LTE |

Tabla 1.1: Frecuencias y bandas LTE en España [8].

### 1.1.2. Hacia las redes IP

Hasta la década de 1970 únicamente existían las redes basadas en la conmutación de circuitos. En ellas se debe establecer un circuito físico entre los medios de comunicación para la posterior conexión entre los usuarios. Dicho circuito permanece activo durante la comunicación, liberándose al terminar; éste es el principio de funcionamiento de la red telefónica básica. Las redes de conmutación de circuitos, entre otras ventajas, permiten la transmisión de datos en tiempo real. Además, cada conexión posee exclusivamente un recurso físico (un circuito) mientras esté activa la conexión, lo que permite que ambas

partes se puedan comunicar a la máxima velocidad limitada por el circuito establecido. Por otro lado, entre las desventajas nos encontramos con el desperdicio de ancho de banda, cuando otros circuitos de los equipos no se están utilizando. También tenemos la incapacidad de reajustar el circuito, por ejemplo, si surgen nodos intermedios con menor costo entre ellos. Y presenta baja tolerancia a fallos ya que, si un nodo se cae, el circuito lo hará con él.

Por estas razones surgió la necesidad de compartir recursos de una forma más eficiente, naciendo así el concepto de conmutación de paquetes. En este tipo de conmutación, la información se divide en paquetes del mismo tamaño. Estos paquetes incluyen una cabecera con la dirección de origen, la de destino y además datos de control, que serán utilizados, por ejemplo, para establecer la *Quality of Service* (QoS). Con este principio, dichas redes ofrecen mayor flexibilidad y rentabilidad de los nodos intermedios, retransmitiendo una cantidad menor de datos en caso de pérdida de información.

El protocolo *Internet Protocol* (IP) está diseñado para la comunicación bidireccional, transmitiendo datos mediante un protocolo no orientado a conexión basado en la conmutación de paquetes. Este protocolo direcciona los paquetes mediante direcciones lógicas de 32 o 64 bits, según la versión,

Las redes LTE emplean un sistema denominado *"all IP"*, ya que utiliza este protocolo en todo el sistema, desde la red de acceso hasta el núcleo de red. Además del tráfico de datos, el tráfico de voz también se transmite sobre este protocolo usando conmutación de paquetes , es decir, usando VoIP (*Voice over IP*). De esta manera se consigue una mejor integración con otros servicios multimedia.

### 1.1.3. Virtualización de equipos

Debido al aumento de equipos específicamente creados para llevar a cabo un objetivo concreto, y gracias a la expansión de las prestaciones de ordenadores de propósito general, surgió la necesidad de simplificar costes, equipos y todo lo que conlleva. Para dar solución a este problema, además de tener la capacidad de detectar errores previamente al lanzamiento de un equipo específico, nació la virtualización de equipos.

Cuando hablamos de virtualización nos referimos a particionar los recursos de un equipo físico entre varios equipos virtuales. Ello permite a cada máquina virtual interactuar de forma independiente con otros dispositivos, aplicaciones, usuarios... , como si de un recurso físico independiente se tratase. Dichas máquinas pueden ejecutar diferentes sistemas operativos, múltiples aplicaciones y utilizar los mismos recursos del equipo físico en el que se encuentran. Estas máquinas están inicialmente aisladas entre ellas por lo que, si ocurriese algún fallo en una de ellas, no afectaría al resto.

Existen multitud de herramientas *software* que hacen posible la virtualización de equipos; éstas asignan los recursos físicos correspondientes a cada una de las máquinas permitiendo al usuario establecer qué recursos y qué cantidades le serán asignados. Algunas de las herramientas más conocidas son: VMware, VirtualBox y VirtualPC.

## 1.2. Motivación

Con el lanzamiento en los últimos años del despliegue de redes de cuarta generación, el aumento en la demanda del acceso a servicios de Internet por parte de usuarios de teléfonos móviles, y la transformación de una sociedad cada vez más involucrada con el campo de las TIC, surge la necesidad de crear soluciones sencillas, abiertas y que se adapten a cualquier situación.

El presente proyecto trata de realizar un estudio de una red celular LTE, basada en el estándar del *Third Generation Partnership Project* (3GPP), con la plataforma *Open Air Interface* (OAI). Así, se pretende desplegar una pequeña red celular propia, estable y que cumpla con los requisitos de la normativa. Esto permitirá la simplificación en la utilización de recursos y, con ello, reducir los costes que puede llevar consigo el despliegue de dicha red.

Gracias a la reducción de costes de los equipos radio definidos por *software* o *Software Defined Radio* (SDR), el aumento de prestaciones de los PCs y la virtualización de equipos, es posible diseñar un servicio de comunicaciones tan complejo como puede ser una red móvil de cuarta generación mediante un equipamiento de bajo coste. Siendo ésta la aproximación que seguiremos en este proyecto.

## 1.3. Objetivos

El principal objetivo de este proyecto es familiarizarse con las redes de cuarta generación (LTE), así como implementar una red móvil con las características descritas por el estándar del 3GPP. Para ello, se utilizará un equipo SDR funcionando como una estación base *evolved Node Base* (eNB), dos móviles comerciales y un PC que funcionará como núcleo de red.

Así, como primer paso realizaremos la emulación de dicho sistema mediante máquinas virtuales. Esto permitirá estudiar su configuración, instalación, y verificación de su funcionamiento. Esta característica nos permite implementar todo un sistema de red LTE en un único PC.

Una vez emulado el sistema, se pondrá a prueba las capacidades del equipo SDR para que funcione como una entidad eNB de una red LTE. El uso de un SDR conlleva una gran reducción de costes a la hora de realizar la implantación de un eNB en una red móvil.

Por último, se realizará un análisis de la capa física y del intercambio de señalización dentro de una red LTE creada a partir de la plataforma libre OAI.

## 1.4. Organización del proyecto

Este apartado trata de mostrar cómo está estructurada la memoria del proyecto, de tal manera que facilite cualquier consulta por el lector, así como poder realizar una búsqueda más eficaz. La presente memoria consta de:

**Capítulo 1: Introducción**

En este capítulo se realiza una breve introducción de la actualidad de las redes móviles, así como los diferentes aspectos que han motivado la realización del proyecto. Finalmente se presenta una recopilación de los objetivos que se plantearon en un principio, y un breve resumen de cada una de las partes en las que se compone el proyecto.

**Capítulo 2: Estado del arte**

En este capítulo se expondrán diferentes soluciones que existen similares a las utilizada en este proyecto, destacando las diferencias entre ellas. Finalmente, se justifican las razones que nos han llevado a seleccionar la plataforma OAI.

**Capítulo 3: Planificación y estimación de costes**

En este capítulo se detalla la planificación temporal del proyecto, así como cada una de las fases en las que se divide. Además, se describen los recursos necesarios para la realización del mismo y, finalmente, se estiman los costes parciales con el objetivo de mostrar el coste total asociado al proyecto.

**Capítulo 4: Resumen del estándar LTE**

En este capítulo se describen conceptos básicos para el comprender el funcionamiento de una red móvil LTE, así como la arquitectura por la que está compuesta. Además, se explican los distintos protocolos que operan entre la *Evolved UMTS Terrestrial Radio Access Network* (E-UTRAN) y el *Evolved Packet Core* (EPC) para su comunicación. Posteriormente, se detallan la capa física y la capa *Medium Access Control* (MAC) para entender cómo se realiza la adecuación de la información para ser enviada a través de la interfaz aire. Finalmente, se detallan los mensajes intercambiados en distintos procesos de un sistema LTE, como es la selección de celda por parte de un usuario o el registro en la red, y los mensajes intercambiados para proporcionar distintos mecanismos de seguridad.

**Capítulo 5: *Open Air Interface***

En este capítulo se describe la plataforma utilizada para el desarrollo del proyecto. Posteriormente, se detallan las partes en las que se divide la plataforma, así como las distintas posibilidades que nos ofrece.

**Capítulo 6: Instalación y configuración de OAI**

En este capítulo describimos la puesta en marcha de los distintos equipos utilizados para la parte práctica del proyecto. Seguidamente, se explican la instalación y configuración llevadas a cabo para la simulación de una red LTE con *Open Air Interface System Emulation* (OAISIM). Finalmente, se detallan la instalación y configuración realizadas para poner en funcionamiento una red LTE con la plataforma OAI, para posteriormente poder analizar su funcionamiento.

**Capítulo 7: Análisis**

En este capítulo tiene lugar el análisis de la red móvil creada. Este estudio comienza en la parte radio, continua con el análisis de trazas de distintos procedimientos como el registro en la red de un UE, y finalmente termina con pruebas de velocidad. Se llevará a cabo una comparativa con los requisitos exigidos por el 3GPP para las redes de cuarta generación.

**Capítulo 8: Conclusiones**

Este capítulo concluye la memoria, exponiendo las conclusiones finales a las que se han llegado una vez terminado el proyecto. Además, se expone una serie de posibles trabajos futuros y se finaliza con una valoración personal del trabajo que se ha llevado a cabo.

**Anexo A. Máquina virtual EPC**

En este anexo se recogen los archivos de configuración usados en la máquina virtual que ejecuta el EPC de nuestra red móvil LTE. Se incluyen los archivos para la configuración de red, el *Home Subscriber Server* (HSS), el *Mobility Management Entity* (MME) y el *Serving Gateway* (S-GW).

**Anexo B. Máquina virtual OAISIM**

En este anexo se recogen los archivos de configuración usados en la máquina virtual que ejecuta OAISIM. Se incluyen los archivos para la configuración de red y el eNB.

**Anexo C. Manual del usuario OAI: CN y eNB**

En este anexo se recoge un breve resumen de los comandos necesarios para la compilación y lanzamiento de las distintas entidades por las que se compone OAI (HSS, MME, SPGW y eNB). Además, se realiza una recopilación de los distintos parámetros opcionales que se pueden incluir en el lanzamiento de cualquiera de éstas entidades.

**Bibliografía**

En la bibliografía se detallan las referencias de las que se ha hecho uso para adquirir la información y los conocimientos para realizar el presente proyecto.

# Capítulo 2

# Estado del Arte

## 2.1. Introducción

En este capítulo se expondrán diferentes soluciones que existen similares a las utilizada en este proyecto, destacando las diferencias entre ellas. Finalmente, se justifican las razones que nos han llevado a seleccionar la plataforma OAI.

## 2.2. Amari LTE 100

### 2.2.1. Introducción

El *software* Amari LTE 100 [12] es un producto comercializado por la empresa francesa Amarisoft. Éste fue creado para estudiantes e investigadores que deseen adquirir una estación base portátil y desplegar así una red LTE.

### 2.2.2. Características

Amari LTE 100 es un *software* cerrado y de pago. Dicho paquete incluye todos los archivos necesarios para realizar la implementación *software* de los principales bloques en los que se compone una red LTE:

- EPC
- eNB
- UE
- Y dos componentes adicionales:
  - MBMS
  - IMS

Ésta suite de *software* nos permite la comunicación de hasta 1000 UE comerciales, y además sus componentes son compatibles con el *Release 13* del 3GPP. Las distintas entidades se configuran a través de archivos de configuración JSON. Por otra parte, el sistema posee de una API llamada *WebSocket* para su puesta en marcha y administración remota.

Otra de las características que nos ha llamado la atención es que las entidades eNB son capaces de soportar la tecnología NB-IoT. Además, este paquete *software* también dispone de un simulador de UE.

### 2.2.3. Ejecutables

Éste *software* dispone de varios ejecutables que corresponden con los principales elementos de una red LTE y son los encargados de implementar virtualmente las entidades que veremos a continuación:

- **LTEMME**: este ejecutable implementa las siguientes entidades que forman el EPC: el MME, el S-GW, el *Packet Data Network Gateway* (P-GW) y el HSS.
- **LTEENB**: este ejecutable implementa la red E-UTRAN, es decir, implementa una estación base eNB.
- **LTEIMS**: habilita el servicio multimedia o *IP Multimedia Subsystem* (IMS), para dar soporte mediante el protocolo IP a los servicios de voz, datos y multimedia.
- **LTEMBMSGW**: este ejecutable se encarga de habilitar el elemento para *Multimedia Broadcast and Multicast System* (MBMS), encargándose de proporcionar una transmisión broadcast y de servicios multicast a las redes móviles 3GPP.

## 2.3. srsLTE

### 2.3.1. Introducción

El *software* srsLTE [13] pertenece a la empresa irlandesa Software Radio Systems Limited (SRS), especializada en *software* de alto rendimiento. Esta empresa ofrece implementaciones modulares y portátiles para una amplia gama de tecnologías inalámbricas.

### 2.3.2. Características

srsLTE es una librería de *software* libre y de código abierto, aunque está disponible bajo licencias comerciales. Está diseñada para aplicaciones SDR e implementa únicamente la parte de red E-UTRAN, virtualizando así la estación base eNB y el UE; por otra parte los archivos de configuración y desarrollo se han realizado en lenguaje C.

Otra de las características a destacar es que utiliza *software* del proyecto OpenLTE para algunas funciones de seguridad y para el análisis de mensajes RRC/*Non-Access Stratum* (NAS). Cumple con la *Release 8* del 3GPP [3] y únicamente permite la configuración para la transmisión en *Frequency Division Duplexing* (FDD). Éste *software* ha sido probado exitosamente con los siguientes equipos: USRP B210, USRP X300, bladeRF y limeSDR.

### 2.3.3. Ejecutables

La librería srsLTE dispone de dos ejecutables, encargados de implementar virtualmente las entidades anteriormente mencionadas. Son:

- **srsUE**: se trata de una aplicación completa para la simulación de un UE, que implementa todas las capas intermedias incluyendo la capa física hasta la capa de red IP.
- **srsENB**: implementa virtualmente una estación base eNB. Está compuesta por módulos que proporcionan el funcionamiento al conjunto de protocolos utilizado entre en eNB y el EPC de la red LTE, como son las capas PHY, MAC, RLC, PDCP, RRC, NAS, S1AP y GW.

## 2.4. Open-LTE

### 2.4.1. Introducción

Open-LTE [2] es una herramienta *software* para realizar la simulación de una red E-UTRAN, aunque implementa un EPC simplificado. Está desarrollada en C++ y Python, y se trata de una solución de código libre.

### 2.4.2. Características

Este *software* implementa la parte de acceso radio, un simulador para el EPC con las funcionalidades del MME y del HSS, incluyendo además herramientas para escanear y grabar señales LTE basadas en GNU Radio. OpenLTE está centrado en la transmisión y recepción del enlace descendente. Esta plataforma incorpora los archivos necesarios para pruebas y simulaciones del enlace descendente con esquema FDD usando 5 MHz para el ancho de banda del canal. La configuración de Open-LTE se realiza a través de un archivo de configuración en formato XML que establece los parámetros del escenario a simular. Nos permite configurar el ancho de banda, la distancia entre el eNB y el UE, la velocidad y dirección de los usuarios, etcétera.

### 2.4.3. Componentes

Open-LTE tiene tres componentes principales:

- El módulo de red virtual LTE que proporciona la simulación de la conectividad IP de la red.
- El *gateway* que proporciona la capacidad de recibir y transmitir paquetes de datos a través del enlace físico.
- La tabla de enrutamiento que nos proporciona el encaminamiento de los datos desde un extremo a otro. OpenLTE funciona de manera transparente en la capa de red, de la misma forma que un *router*.

## 2.5. Open Air Interface

### 2.5.1. Introducción

OAI es una plataforma de *software* libre, basada en el estándar 3GPP. Proporciona un *software* para la red troncal denominada EPC, y otra parte para la red de acceso. La plataforma está desarrollada en lenguaje C, y se distribuye bajo una licencia libre.

### 2.5.2. Características

Tal y como se ha mencionado anteriormente, la plataforma OAI está dividida en dos partes:

- OpenairCN: implementa los diferentes equipos que forman la red troncal de LTE, y son: MME, HSS, S-GW y P-GW.
- Openair5G: implementa la red de acceso, implementando los eNB. Además, es capaz de implementar distintos usuarios de la red UE.

Este *software* nos ofrece un amplio abanico de posibilidades a la hora de montar el escenario deseado, proporcionando un alto grado de flexibilidad.

En el capítulo 5 se explica con más detalle la plataforma OAI, ya que ha sido la seleccionada para llevar a cabo el presente proyecto. En él se explican sus características, las posibilidades que nos ofrece para evaluar distintos escenarios, así como las partes en las que está compuesto.

## 2.6. Conclusiones

Como resultado de las distintas soluciones para realizar el presente proyecto, hemos de decir que la primera de ellas, Amari LTE 100 tiene por contra que se trata de una solución de pago aunque es la más completa de todas ya que ofrece multitud de configuraciones y posibilidades. Las otras dos soluciones están más orientadas a la simulación y emulación que a la experimentación real con dispositivos. Además, no nos permiten poner en funcionamiento una red completa LTE, debido a que están centradas en la parte de red E-UTRAN. Aunque estas dos son de código abierto, srsLTE es una herramienta más estable, con menos puntos débiles y más sofisticada que Open-LTE. Finalmente, hemos de decir que Open-LTE actualmente es una solución incompleta y muchas de sus características no son estables ya que se encuentran en desarrollo. Además, no proporciona una implementación virtual de un UE como tal.

Por estos detalles, hemos escogido OAI como plataforma de desarrollo para este proyecto. Los principales puntos por los que se ha escogido OAI son los siguientes: se trata de un *software* libre, está escrita en C, implementa la red troncal y la red de acceso de una red móvil LTE, proporciona flexibilidad para la evaluación de distintos escenarios.

# Capítulo 3

# Planificación y estimación de costes

## 3.1. Introducción

En el presente capítulo se describe una planificación del proyecto llevado a cabo y una estimación de costes para poder abordar dicho proyecto en el ámbito laboral.

## 3.2. Planificación

En esta sección se ha realizado la descripción de cada una de las etapas en las que se divide este proyecto. El objetivo es realizar una planificación inicial del desarrollo esperado para este proyecto.

Las diferentes partes en las que se divide la planificación son:

- **Estudio de la tecnología LTE.**

  En primer lugar, procederemos a la búsqueda de información sobre la tecnología LTE. Analizaremos y estudiaremos su arquitectura, los principales componentes que la forman y los protocolos utilizados en cada una de las partes. Además, deberemos investigar qué características la destacan de las generaciones anteriores y qué especificaciones son necesarias para una red móvil LTE.

- **Soluciones de mercado.**

  En segundo lugar, realizaremos una búsqueda de información acerca de las soluciones que actualmente están disponibles en el mercado, independientemente de que sean soluciones abiertas o cerradas. Destacaremos sus características y cuál es su estructura.

- **Acerca de *Open Air Interface*.**

  En tercer lugar, se procederá a recopilar la máxima información posible acerca de la puesta en marcha del proyecto para así interiorizar la información y comprender el funcionamiento de la tecnología implicada. El uso de esta herramienta es un requisito inicial dado por los tutores.

- **Estudio de software de virtualización.**

  Posteriormente, se investigará acerca del software de virtualización. En este proyecto nos hemos decantado por *VMware Workstation 12 Player*, si bien se podrían utilizar otros gestores de virtualización como VirtualBox. La virtualización permitirá implementar un escenario de pruebas en un único ordenador y comprobar

el correcto funcionamiento de la configuración final, que posteriormente será implementada en los equipos reales.

- **Primeros pasos con OAI**

  En cuarto lugar, recopilaremos información para facilitar la utilización de la herramienta *Open Air Interface*. Además, es necesario conocer las distintas posibilidades que nos ofrece, para poder escoger la más indicada a nuestras necesidades. Estudiaremos su arquitectura y los archivos de configuración e instalació.

- **Simulación en máquinas virtuales.**

  Una vez que tengamos los conocimientos necesarios, procederemos a realizar la instalación y la configuración de los distintos dispositivos que nos permiten utilizar OAISIM en equipos virtuales. Esta etapa es quizá la más costosa y complicada ya que, una vez comprobado su correcto funcionamiento, nos resultará más sencillo trasladar dicha configuración a los equipos reales y continuar con el análisis en la red móvil LTE que pondremos en funcionamiento.

- **Preparación del escenario real.**

  Seguidamente, una vez completada la simulación virtual del escenario, es necesario trasladarlo a los equipos reales. Primero necesitamos conocer las especificaciones técnicas de los equipos disponibles para poder realizar los ajustes necesarios, como pueden ser la conexión entre ellos, la conexión a Internet, desactivar estados en la BIOS que perjudiquen el rendimiento de nuestro PC, etcétera.

- **Instalación y configuración de OAI.**

  Una vez que los equipos se encuentran con la configuración adecuada, realizaremos la instalación y configuración de OAI. Además, realizaremos la configuración de las tarjetas USIM programables para que los UE puedan conectarse a la red. Además, habrá que introducir los datos necesarios en la base de datos que utilizará la entidad HSS. Estos últimos pasos son necesarios ya que la autenticación, mútua entre UE y EPC, es obligatoria en las redes LTE.

- **Test y análisis**

  Finalizada la implementación de OAI en los equipos reales, se procederá a la realización de una serie de pruebas básicas. En esta parte se comprobará la sincronización de los equipos, la transmisión de información en la banda previamente establecida, la autenticación y acceso a la red por parte del usuario, el acceso a Internet, el análisis del enlace físico, etcétera.

- **Elaboración de la memoria técnica.**

  Finalmente, se documentará toda la información acerca del trabajo realizado, recopilando los aspectos teóricos y técnicos, así como las implementaciones realizadas de forma detallada. Además, se recogerán los resultados obtenidos del análisis del sistema una vez puesto en funcionamiento.

Una vez identificadas todas las partes en las que se divide el proyecto, realizaremos una planificación con un diagrama de Gantt. Este diagrama con la planificación inicial tiene carácter orientativo, ya que existen numerosos factores que influyen a la hora de realizar el proyecto, aunque la idea original es aproximar el desarrollo del proyecto a los tiempos marcados en cada plazo.

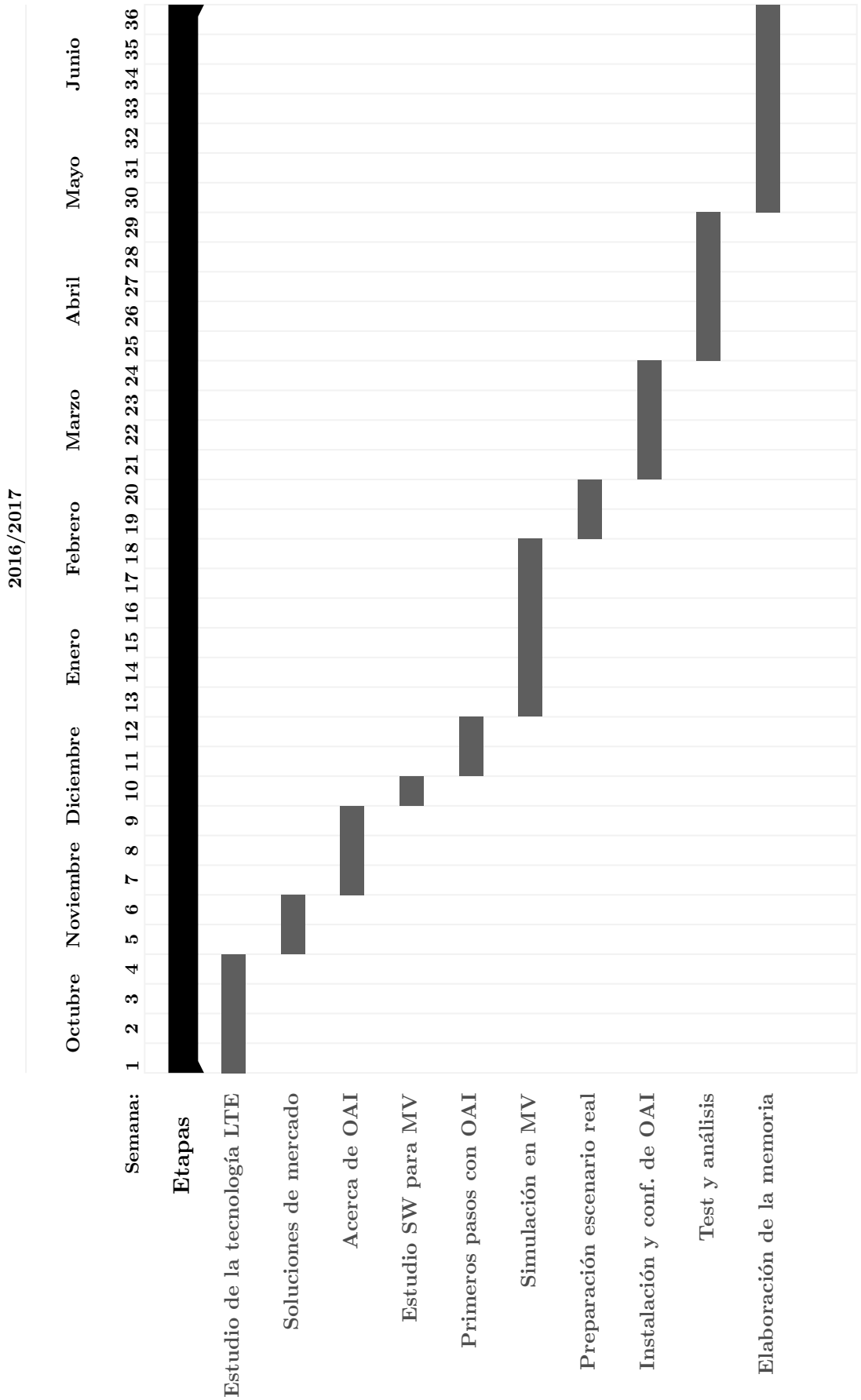



Figura 3.1: Diagrama de Gantt del proyecto.

Por otra parte, se ha realizado el desglose según las horas de trabajo planificadas inicialmente en cada una de las partes. En la tabla 3.1 tenemos la distribución inicial estimada.

| Etapa | Descripción | Tiempo (h) |
|---|---|---|
| 1 | Estudio de la tecnología LTE | 40 |
| 2 | Soluciones de mercado | 20 |
| 3 | Acerca de OAI | 30 |
| 4 | Estudio SW para MV | 10 |
| 5 | Primeros pasos con OAI | 20 |
| 6 | Simulación en MV | 60 |
| 7 | Preparación del escenario real | 20 |
| 8 | Instalación y conf. de OAI | 40 |
| 9 | Test y análisis | 50 |
| 10 | Elaboración de la memoria | 70 |
| | TOTAL | 360 horas |

Tabla 3.1: Distribución temporal de proyecto.

## 3.3. Recursos utilizados

En esta sección identificaremos todos los recursos empleados en el proyecto. Se ha realizado una clasificación en tres tipos de recursos: hardware, software y humanos.

### 3.3.1. Recursos *hardware*

#### PC de sobremesa Hiditec

Se ha empleado un PC de sobremesa para poner la puesta en funcionamiento del escenario real. Este equipo albergará al EPC ($HSS + MME + S/P - GW$) y el eNB. Conectaremos mediante un cable USB 3.0 el dispositivo SDR, y también mediante este PC realizaremos la lectura y programación de las tarjetas *Universal Subscriber Identity Module* (USIM) utilizadas. Las principales características de este equipo se detallan a continuación:

| | | |
|---|---|---|
| General | Marca | Hiditec |
| | Modelo | Q6 PSU 500 |
| Tamaño | Profundidad | 172 mm |
| | Altura | 416 mm |
| | Ancho | 365 mm |
| | Peso | 4.215 kg |
| Componentes | Procesador | Intel Core i7-4790 |
| | Velocidad del procesador | 3.60 / 4.00 GHz Turbo, Caché 8 MB |
| | HDD Capacidad de disco | 500GB |
| | Memoria RAM | 16GB |
| | Unidad óptica | DVD-Super (DVD±R/±RW/RAM/±R9) |
| | Puertos USB | 5 |
| Otros | Sistema operativo | Ubuntu 17.04, 64 bits |

Tabla 3.2: Características del PC Hiditec.


Figura 3.2: PC de sobremesa Hiditec.

### USRP National Instruments mod.2901

Se utiliza la tecnología SDR para implementar la estación base LTE. Concretamente hemos utilizado un dispositivo *Universal Software Radio Peripheral* (USRP)de la marca National Instruments modelo 2901 [11], que se puede ver en la figura 3.3.Sus principales características son las siguientes:

- Frecuencia central ajustable desde 70 MHz hasta 6 GHz.
- Soporta tecnología MIMO 2X2.
- Razón máxima de transferencia de 15 MS/s
- Precisión en frecuencia 2.5 ppm
- Máximo ancho de banda 56 MHz
- Figura de ruido 5 dB a 7dB

Figura 3.3: USRP NI 2901.

### Antena Siretta

Las antenas que hemos utilizado para dotar al USRP de conectividad 4G, han sido antenas pertenecientes a la marca Siretta modelo Delta 1A/-x/SMAM/S/RA/11. Se trata de una antena cuatribanda que es capaz de ajustarse a las frecuencias de GSM/GPRS, 3G/UTMS, ISM (868/915 MHz), 4G.

Las especificaciones de este tipo de antenas son:

- Temperatura de funcionamiento: -20 a +60 ºC
- Impedancia: 50Ω
- Ganancia: 2 dBi (3,5 dBi de pico)
- VSWR: 2 máx.
- Polarización: vertical
- Conector: SMA macho

Figura 3.4: Antena Siretta Delta 1A.

### PC portátil - Toshiba L850

Emplearemos un ordenador portátil para la creación de las máquinas virtuales para la simulación del proyecto con OAISIM. Además, se utilizará para todo lo relacionado con búsqueda de información, y desarrollo de la presente memoria. El PC que se ha usado se muestra en la figura 3.5, cuyas características técnicas se muestran especificadas en la tabla 3.3.


Figura 3.5: PC Toshiba L850-1UX [16].

| | | |
|---|---|---|
| General | Marca | Toshiba |
| | Familia | Satellite |
| | Modelo | L850-1UX |
| Tamaño | Profundidad | 33.5 mm |
| | Altura | 380 mm |
| | Ancho | 242 mm |
| | Peso | 2.4 kg |
| Componentes | Procesador | Intel Core i7-3630QM |
| | Velocidad del procesador | 2.40/3.40GHz Turbo, 1.6MHz FSB, 6MB L3 |
| | Tarjetas Gráficas | AMD Radeon HD 7670M de 2 GB DDR3 |
| | HDD Capacidad de disco | 500GB |
| | Memoria RAM | 12GB |
| | Unidad óptica | DVD (DVD±R/±RW/RAM/±R9) |
| | Puertos HDMI | 1 |
| | Puertos USB | 3 |
| Display | Tamaño de pantalla | 15,6” |
| | Tipo de pantalla | LED |
| Otros | Sistema operativo | Windows 10 |
| | Versión del Bluetooth | 4.0 |
| | WiFi | Si |

Tabla 3.3: Características del PC Toshiba L850.

### Terminal Móvil - BQ X5 Plus

Uno de los terminales móviles que se ha utilizado para la realización de test y análisis en la implementación de la red LTE es el X5 Plus fabricado por la empresa española BQ [6].


Figura 3.6: BQ X5 Plus

Sus características se muestran en la siguiente tabla, 3.4:

| | | |
|---|---|---|
| General | Marca | BQ |
| | Modelo | X5 Plus |
| Tamaño | Dimensiones | 70x145x7,7 mm |
| | Peso | 145 g |
| Pantalla | Dimensiones | 5” |
| | Tecnología | LCD IPS LTPS Multitáctil, GFF |
| | | Cristal $Dinorex^{TM}$(NEG) Quantum Color+ |
| | Resolución | Full HD 1080 x 1920 - 440 ppi |
| | Relación de aspecto | 16:9 |
| | Brillo | 620 nits |
| Procesador | CPU | Qualcomm $Snapdragon^{TM}$ 652 Octa Core 1,8GHz(PoP) |
| | GPU | Qualcomm $Adreno^{TM}$ 510 hasta 600 MHz |
| Interfaz | Sistema Operativo | Android 7.1.1 Nougat |
| Memoria | Interna | 32 GB |
| | RAM | 3 GB |
| Energía | Batería | LiPo 3200 mAh |
| Cámara | Trasera | Sony IMX298, 16 MP |
| | Frontal | Sony IMX219, 8 MP |
| Conectividad | Red | 4G (LTE) FDD (800/1800/2100/2600)MHz-(B20/B3/B1/B7) |
| | | 3G HSPA+ (850 / 900/ 1900 / 2100)MHz-(B5/B8/B2/B1) |
| | | 2G GSM (850/900/1800/1900)MHz |
| | Posicionamiento | GPS + GLONASS + GALILEO |
| | Wi-Fi | 802.11b/g/n/ac (dual band - 2,4 GHz / 5 GHz) |
| | Bluetooth | 4.2 |
| | NFC | (HCE+SE) |

Tabla 3.4: Características del terminal móvil BQ X5 Plus.

### Terminal Móvil - Xiaoimi Mi5


Figura 3.7: Xiaomi Mi5

El segundo terminal móvil que hemos utilizado para las distintas pruebas en la implementación de nuestra red LTE ha sido el modelo Mi5 de la empresa china Xiaomi [17]. Podemos ver el modelo en la figura 3.7, y sus características técnicas en la tabla 3.5.

| | | |
|---|---|---|
| General | Marca | Xiaomi |
| | Modelo | Mi5 |
| Tamaño | Dimensiones | 69,2x144,55x7,25 mm |
| | Peso | 129 g |
| Pantalla | Dimensiones | 5,15” |
| | Tecnología | LCD IPS Multitáctil |
| | Resolución | Full HD 1080 x 1920 - 428 ppi |
| | Relación de aspecto | 16:9 |
| | Brillo | 600 nits |
| Procesador | CPU | Qualcomm $Snapdragon^{TM}$ 820 Octa Core 1,8GHz (PoP) |
| | GPU | Adreno 530 |
| Interfaz | Sistema Operativo | Android 6.0 Marshmallow |
| Memoria | Interna | 32 GB |
| | RAM | 3 GB |
| Energía | Batería | LiPo 3000 mAh |
| Cámara | Trasera | 16 MP |
| | Frontal | 4 MP |
| Conectividad | Red | LTE FDD: (850/1800/2100/2600)MHz -(B5/B3/B1/B7) |
| | | LTE TDD: (1900/2300/2500/2600)MHz-(B39/B40/B41/B38) |
| | | TD-SCDMA: 1900/2000 MHz |
| | | WCDMA: 850/900/1900/2100 MHz |
| | | GSM: 850/900/1800/1900 MHz |
| | Posicionamiento | GPS + AGPS + GLONASS + BeiDou |
| | Wi-Fi | 802.11a/b/g/n/ac (dual band - 2,4 GHz / 5 GHz) |
| | Bluetooth | 4.2, HID |
| | NFC | Sí |

Tabla 3.5: Características del terminal móvil Xiaomi Mi5.

### Tarjetas USIM - SYSMOCOM

Uno de los dispositivos imprescindibles para poder conectarnos y realizar nuestra autenticación en una red LTE es la tarjeta USIM. En el presente proyecto se han utilizado tarjetas USIM programables de la marca SYSMOCOM [15].


Figura 3.8: Tarjeta USIM de SYSMOCOM.

Para poder realizar su programación con los valores de parámetros específicos, se ha utilizado un lector de tarjetas (véase la figura 3.9), desarrollado por la compañía Gemalto [9].

Figura 3.9: Lector de tarjetas Gemalto GemPC Twin IDBridge CT30.

### Analizador de espectros - N9010A

Uno de los dispositivos que nos ayudarán a descubrir problemas y ver si nuestra estación base cumple con las especificaciones del interfaz radio es el analizador de espectros. Disponemos del analizador de espectros EXA N9010A de la marca AGILENT (véase la figura 3.10).

Este analizador de espectros tiene instaladas aplicaciones de propósito general para analizar la parte radio de nuestra red LTE. Así, podremos analizar el ancho de banda efectivo tanto del enlace descendente como el descendente; además, podremos obtener datos y gráficas que nos mostrarán la densidad espectral de potencia de la banda en la que estaremos transmitiendo. Por otra parte, dispone de aplicaciones específicas para el análisis de redes celulares, en nuestro caso para LTE. Así, será posible obtener la constelación de la señal transmitida, la magnitud del vector de error o *Error Vector Magnitude* (EVM), así como la representación de la señal en el dominio de la frecuencia y en el dominio del tiempo.


Figura 3.10: Analizador de espectros AGILENT EXA N9010A [10].

### 3.3.2. Recursos *software*

- Sistema operativo *Windows 10 Home* (64 bits), instalado en el ordenador portátil, sobre el que instalaremos las máquinas virtuales.
- Sistema operativo *Linux Ubuntu 16.04.04* (64 bits), instalado en las máquinas virtuales. Una de ellas es utilizada para la instalación del EPC de OAI, mientras que la otra se instala y configura la parte E-UTRAN.
- *OAI-CN.* Es una herramienta de software libre, parte del paquete OpenAirInterface, que permite la instalación del núcleo de red ó EPC de una red LTE, mediante la cual podemos instalar y configurar el HSS, MME, S-GW y el P-GW.
- *OAISIM.* Es una herramienta, parte del paquete OpenAirInterface, que permite la simulación y emulación de una red OpenAirLTE. Proporciona una simulación de la capa física completa y canales radio sintéticos (simulados en detalle) o con una abstracción de la capa física (con una carga computacional menor). Nos permite virtualizar varios nodos eNB y UE en el mismo proceso. También proporciona soporte para la emulación basada en Ethernet para que los nodos se puedan distribuir en una red de PCs.
- *VMware.* Es una herramienta para la virtualización de máquinas, utilizada para crear el escenario virtual del proyecto.
- *MySQL.* Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation. Se utiliza para almacenar los datos de las tarjetas USIM programables.
- *Wireshark.* Es un analizador de protocolos de software libre, utilizado para capturar los paquetes procesados por el EPC.
- *Software 89600 Vector Signal Analyzer.* Es un conjunto completo de herramientas para la demodulación y el análisis vectorial de señales. Con él obtendremos, además de las constelaciones utilizadas en las señales o el conjunto de slots temporales, parámetros tan importantes como el EVM.
- *MATLAB.* Es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).
- *Toolbox de LTE.* Esta toolbox es una extensión para MATLAB que está destinada para diseñar y comprobar sistemas inalámbricos según el estándar LTE.
- *Overleaf.* Es una herramienta colaborativa en la nube basada en LaTeX. Con ella ha sido posible la elaboración de la memoria del proyecto, permitiendo así un seguimiento por parte de los tutores del proyecto.

### 3.3.3. Recursos humanos

- D. Jorge Navarro Ortiz, Profesor Contratado Doctor en el Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.
- D. José Carlos Segura Luna, Catedrático de Universidad en el Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.

- Francisco García Espigares, alumno del Grado de Ingeniería de Tecnologías de Telecomunicación y autor del presente proyecto.

## 3.4. Costes

En esta sección trataremos de reflejar la estimación de costes de este proyecto.

### Recursos *hardware*

Los costes asociados a los recursos *hardware* que han sido listados en el apartado 3.3.1 se detallan en la tabla 3.6.

| Descripción | Coste (€) | Vida media | Utilización | SubTotal (€) |
|---|---|---|---|---|
| PC sobremesa | 1.200,00 | 36 meses | 10 meses | 333,33 |
| USRP B210 | 1.576,00 | 36 meses | - | 1.576,00 |
| Antena Siretta | 8,27 | 36 meses | - | 16,54 |
| PC Toshiba | 750,00 | 36 meses | 10 meses | 208,33 |
| Móvil BQ X5 Plus | 315,00 | 24 meses | - | 315,00 |
| Móvil Xiaomi Mi5 | 297,00 | 24 meses | - | 297,00 |
| USIM SYSMOCOM | 70,00 | 24 meses | - | 70,00 |
| AGILENT EXA N9010A | 18.228,00 | 36 meses | 3 meses | 1.519,00 |
| Línea de acceso a Internet | 30 | - | 10 meses | 300,00 |

Tabla 3.6: Costes de recursos *hardware*

### Recursos *software*

La mayoría de los recursos *software* son gratuitos, por lo que conlleva una importante reducción en los costes del presupuesto final del proyecto, pero algunos de ellos son software propietario y/o comercial. Se detallan en la tabla 3.7.

| Descripción | Coste (€) | Vida media | Utilización | SubTotal (€) |
|---|---|---|---|---|
| Windows 10 Home | 135 | 36 meses | 10 meses | 37,50 |
| MATLAB | 119 | 36 meses | 10 meses | 33,05 |

Tabla 3.7: Costes de recursos *software*

### Recursos humanos

Por otra parte, plasmaremos los recursos humanos en la tabla 3.8. Para ello, se tiene en cuenta el tiempo empleado en cada una de las partes en las que se ha divido la planificación del presente proyecto, como puede verse en la tabla 3.1. Además, y diferenciando el coste medio de los componentes que han intervenido en alguna ocasión en el proyecto, hemos tenido en cuenta los siguientes supuestos:

- Profesor Contratado Doctor de la Universidad de Granada, su coste medio se establece en torno a 50 €/hora.
- Catedrático de la Universidad de Granada, su coste medio se establece en torno a 70 €/hora.

- Graduado en Ingeniería en Tecnologías de Telecomunicación, su coste medio es de 20 €/hora.

A continuación, se presenta el presupuesto teniendo en cuenta las premisas anteriores.

| **Componente** | **Tiempo (h)** | **Coste/h (€)** | **Subtotal (€)** |
|---|---|---|---|
| Jorge Navarro Ortiz | 20 | 50 | 1.000,00 |
| José Carlos Segura Luna | 20 | 70 | 1.400,00 |
| Fco. García Espigares | 360 | 20 | 7.200,00 |

Tabla 3.8: Coste de recursos humanos.

## 3.5. Presupuesto Final

Finalmente, como un resumen global de los diferentes recursos que han sido utilizados en este proyecto, y con el objetivo de presentar una estimación final para llevar a cabo dicho trabajo, en la tabla 3.9 se recogen los costes asociados.

| Concepto | Coste |
|---|---|
| Recursos hardware | 4.635,20 |
| Recursos software | 70,55 |
| Recursos humanos | 9.600,00 |
| **Total** | **14.305,75** |

Tabla 3.9: Presupuesto final.

# Capítulo 4

# Resumen del estándar LTE

## 4.1. Introducción

En el presente capítulo, realizaremos un resumen de la tecnología LTE. Partiremos desde una visión global e iremos desglosando las partes en las que podemos dividir su arquitectura, los dispositivos de los que se compone, los distintos protocolos que utiliza, los canales utilizados para el intercambio de información tanto en el enlace ascendente como en el descendente, y finalmente la composición de la capa física. Para profundizar en el conocimiento de la tecnología LTE se remite al lector a las referencias [18] y [19].

## 4.2. Arquitectura LTE

LTE presenta una arquitectura simplificada en comparación con tecnologías anteriores, ya que está separada en dos bloques bien diferenciados: el núcleo de red o *Core Network* (CN), y la red de acceso radio o *Radio Access Network* (RAN). Los nombres con los que se les conoce comúnmente a dichos bloques en LTE son EPC, para el núcleo de red, y E-UTRAN para la red de acceso radio.

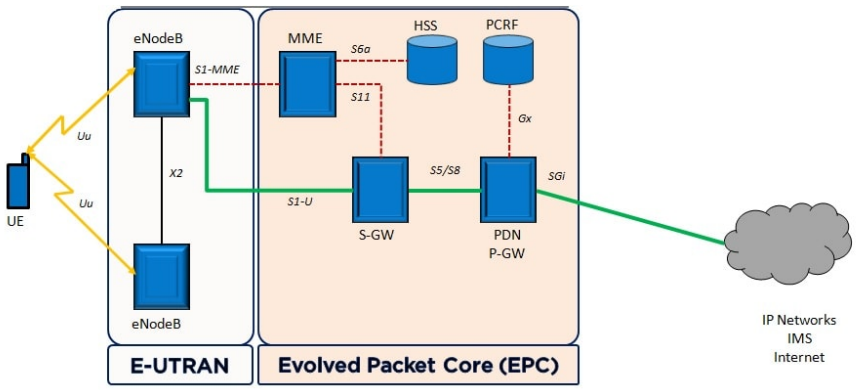



Figura 4.1: Arquitectura LTE [14].

En las siguientes secciones explicaremos detalladamente los elementos que componen cada parte y sus funcionalidades.

### 4.2.1. Visión general

El objetivo de diseño para la arquitectura LTE es minimizar el número de nodos, buscando una solución donde la *Radio Access Network* (RAN) esté formada por un solo nodo y el CN sea lo más independiente posible de ella.

Esta filosofía de reducir el número de equipos ha conducido a la implantación de un nodo más complejo que el NodeB de UMTS, llamado *evolved Node Base* (eNB). Un eNB tiene como función principal la gestión de los recursos radio y la conexión de los terminales móviles a la red.

Por otra parte, el núcleo de red está basado en el núcleo de red del sistema UMTS, y es conocido como EPC. Aquí también se ha implementado la filosofía de minimizar el número de nodos, por lo que posee un nodo que engloba dos entidades funcionales: la entidad de control de la movilidad o MME y el S-GW, más un nodo de enrutamiento a redes externas conocido como *Packet Data Network Gateway* (PDN-GW). La entidad MME es responsable del plano de control, mientras que el S-GW se encarga del plano de usuario o del encaminamiento de los datos.

### 4.2.2. EPC

En esta sección se pretende dar una visión general del EPC, de las entidades que lo componen y sus funcionalidades.

El núcleo de red, aparte de realizar funciones propias del CN como son la facturación, *roaming*, interconexión con redes externas, etcétera, incorpora las siguientes funciones:

- De transporte: soportan la transmisión de información de tráfico y señalización.
- De inteligencia: llevan a cabo aspectos de encaminamiento, gestión de movilidad, control de los servicios ofertados y la calidad de éstos.

El corazón de LTE es el EPC, que representa la evolución de las redes de generaciones anteriores cuyo objetivo es hacer desaparecer la conmutación de circuitos. Para ello implementa un único dominio de paquetes basado en la arquitectura TCP/IP, dando lugar a la solución llamada *all IP*, o todo IP.

La arquitectura del núcleo de red de LTE se puede apreciar en la figura 4.2, en la que aparecen las entidades más importantes así como las interfaces que las conectan.

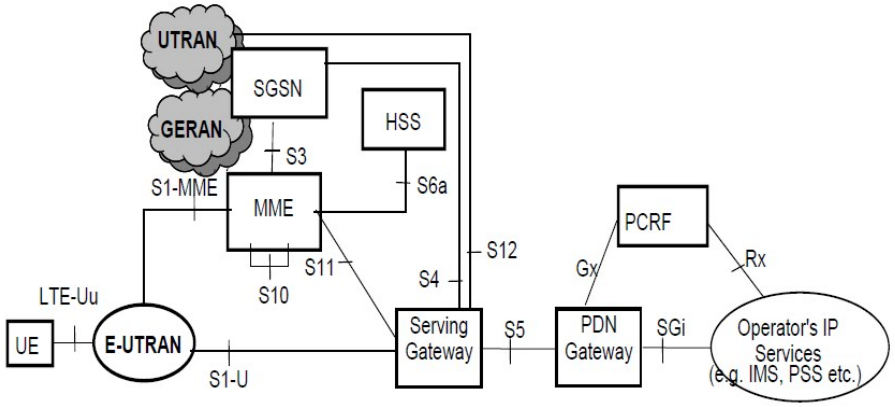



Figura 4.2: Arquitectura del EPC [26]

### HSS

El HSS es una base de datos central que presta servicio a todas las entidades de control que gestionan sesiones y llamadas. Es el responsable de mantener la información relacionada con un usuario como puede ser:

- Identificación de usuario, numeración e información IP.
- Información de seguridad para el control de acceso a la red, autenticación y autorización del usuario.
- Información de ubicación de usuario a nivel de inter-sistema.
- Información del perfil del usuario.

Por otra parte, el HSS proporciona funcionalidad multimedia IP para dar soporte a las funciones de control del IMS. Además, engloba las funciones que anteriormente se llevaban a cabo en el *Home Location Register* (HLR) y en el centro de autenticación o *Authentication Centre* (AuC), las cuales se encontraban en equipos diferentes en redes *General Packet Ratio Service* (GPRS) ó UMTS. Por lo que también genera información de seguridad del usuario para la autenticación mutua, la comprobación de integridad en las comunicaciones y da soporte de cifrado.

### EIR

El equipo de registro de identidad o *Equipment Identity Register* (EIR) es la entidad lógica que se encarga de almacenar las identidades internacionales de los equipos móviles o *International Mobile Equipment Identity* (IMEI) en distintas listas. Éstas indican las condiciones de operación de los UE, estando clasificados en una de las siguientes tres listas disponibles:

- Blanca: el equipo opera con total normalidad.
- Gris: existe alguna restrincción asociada al terminal.
- Negra: estos equipos tiene prohibido el acceso a la red.

### MME

La entidad de gestión de movilidad o MME es la encargada del plano de control dentro del núcleo de red. Las funcionalidades que esta entidad posee son:

- Señalización y seguridad NAS: es la señalización existente entre el terminal móvil y el núcleo de red para diversos procesos como puede ser la conexión a la red o la autenticación.
- Señalización para movilidad entre redes de acceso 3GPP.
- Gestión de las listas de seguimiento.
- Selección de los nodos PDN-GW y del S-GW.
- Selección de *Serving GPRS Support Node* (SGSN) para transferencias a redes de acceso *Segunda Generación Móvil* (2G) o *Tercera Generación Móvil* (3G).
- Itinerancia.
- Funciones de autenticación e intercambio de claves entre el usuario y el núcleo de red.

### S-GW

El nodo S-GW es el principal punto del EPC para el enrutamiento y encaminamiento de paquetes. Es responsable de los *handovers* entre eNB vecinos. Además, se ocupa de la movilidad a otras redes como 2G o 3G, y supervisa y mantiene la información relacionada con la movilidad del usuario durante su estado inactivo. Las principales funciones de esta entidad son las siguientes:

- Encaminamiento de datos desde la red radio hasta el EPC.
- Es un punto de anclaje de encaminamiento para la movilidad, ya sea entre eNB o entre redes distintas de tipo 2G o 3G.
- Almacenamiento de paquetes descendentes enviados a un terminal móvil hasta que se termine el establecimiento de la portadora radio.
- Actúa como filtro para paquetes potencialmente peligrosos para evitar accesos no autorizados al EPC.

### PDN-GW

El nodo PDN-GW es responsable de actuar como un punto fijo de la movilidad entre tecnologías 3GPP y no 3GPP, es decir, permite el acceso a redes como Internet. Esta entidad almacena dos tipos de información, la localización del usuario y la identidad del terminal móvil, y las direcciones *Packet Data Protocol* (PDP). Las funciones más interesantes de esta entidad son las siguientes:

- Filtrado de paquetes para evitar actividades fraudulentas.
- Marcado de paquetes a nivel de transporte en el enlace ascendente y descendente para asegurar las distintas políticas y prioridades de flujo.
- Interceptación legal.

- Asignación de direcciones IP al usuario.
- Encaminamiento de paquetes desde el núcleo de red hasta otras redes externas.
- Tarificación del servicio *Up Link* (UL) y *Down Link* (DL), por ejemplo, basándose en *Service Data Flows* (SDF) definidos por el *Policy and Charging Rules Funcion* (PCRF)

#### PCRF

La entidad PCRF es responsable de la política y control de cargos, decide la forma de manejar los servicios en términos de QoS, y proporciona información al PDN-GW y al nodo S-GW para establecer las políticas de asignación de recursos y políticas de calidad adecuadas. Incluye a dos entidades independientes que son: la *Policy Decision Function* (PDF) y la entidad para la función de reglas de tarificación o *Charging Rules Function* (CRF).

La entidad PDF toma la decisión en la negociación de los requisitos de establecimiento de una conexión, en función de las políticas implementadas. Debe comprobar que los recursos solicitados no exceden los máximos autorizados al usuario para así permitir o rechazar la petición del servicio. Además, deberá definir un contexto PDP.

Por otro lado, el nodo CRF es el encargado de asignar una regla de tarificación previamente definida a cada flujo de datos. Las dos entidades que forman el PCRF intercambiarán información con la entidad PDN-GW para filtrar el tráfico transmitido, asegurar que se cumplen las políticas de servicio y llevar a cabo las políticas de tarificación.

### 4.2.3. E-UTRAN

La arquitectura de red *Evolved Packet Core* (EPC) está formada por eNB conectados entre sí. Tiene una estructura de red plana, por lo que los eNB pueden comunicarse directamente entre sí, sin tener que realizar la señalización y el envío de mensajes a través de un nodo central; esta decisión simplifica la red y es un punto característico en comparación con redes anteriormente estandarizadas por el 3GPP.

La red E-UTRAN lleva a cabo las siguientes funciones:

- Gestión de recursos radioeléctricos.
- Compresión de cabeceras de los paquetes de IP.
- Encriptación de toda la información transmitida por la interfaz radio.
- Transmisión de la señalización hacia el MME, y de los datos del usuario a través del S-GW.

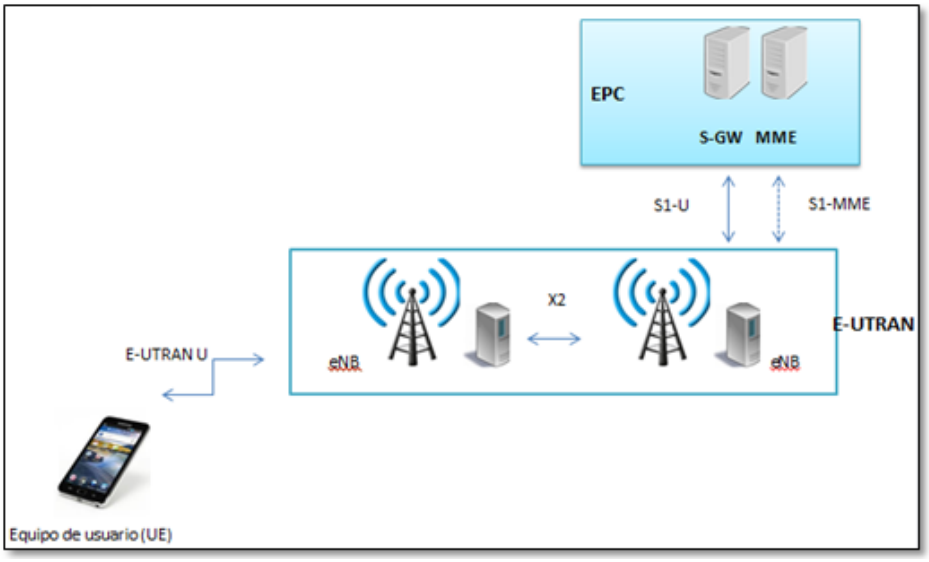



Figura 4.3: Arquitectura de la E-UTRAN [29]

**eNB**

El *evolved Node Base* (eNB) constituye la estación base de la E-UTRAN, por lo que integra toda la funcionalidad de la red de acceso. A diferencia con las estaciones base de tecnologías anteriores como puede ser *Global System for Mobile Communication* (GSM) y UMTS, además de incorporar las funcionalidades de éstas, añade las funcionalidades de los equipos controladores *Base Station Controller* (BSC) y *Radio Network Controller* (RNC).

El eNB es el que realiza todas las funciones que ofrece la E-UTRAN. Al carecer de una unidad central para combinar toda la información de los UE (contexto del UE) entre eNBs junto con los datos temporales almacenados, se ha definido la interfaz X2 evitando así pérdidas en la transmisión de la información.

Se encarga de la transmisión de los paquetes IP hacia y desde los equipos del usuario hasta el núcleo de red EPC, en los que se envía la información de dicho usuario y los mensajes de señalización. Las funcionalidades más características de este nodo son:

- Transmisión de paquetes desde el UE hasta el S-GW.
- Gestión de recursos radio.
- Funciones para el control de admisión.
- Funciones para el control de movilidad.
- Control de interferencias.

Por otra parte, estos equipos son capaces de seleccionar dinámicamente la entidad MME de la red troncal cuando un terminal se registra en la red LTE. Incluso tiene la capacidad de poder conectarse a un conjunto de MME/S-GW. Con esta opción nos permite balancear la carga de señalización y datos del usuario, aumentando así la robustez del sistema. Esta es una de las características de la interfaz S1.

Estos nodos pueden conectarse entre sí sin necesidad de un nodo intermedio, por lo que permiten una gestión más eficiente del uso de los recursos radio así como la reducción de tráfico cuando se produce un handover.


Figura 4.4: eNodeB de Teltronic [27]

## 4.3. Protocolos e Interfaces

En esta sección veremos las arquitecturas de protocolos que encontramos en LTE para la correcta comunicación entre el EPC y la red E-UTRAN; además, haremos referencia a las interfaces que interconectan los equipos mencionados en las secciones anteriores.

Así, explicaremos las distintas capas que componen las arquitecturas sin llegar a exponer la capa física, ya que ésta la veremos más adelante en detalle. En este caso nos centraremos en la arquitectura del protocolo radio que es la que nos interesa para llevar a cabo nuestro proyecto. Según la estandarización de LTE, la arquitectura del protocolo radio se puede separar en el plano de control y el plano de usuario, tal y como se muestra en la figura 4.5.

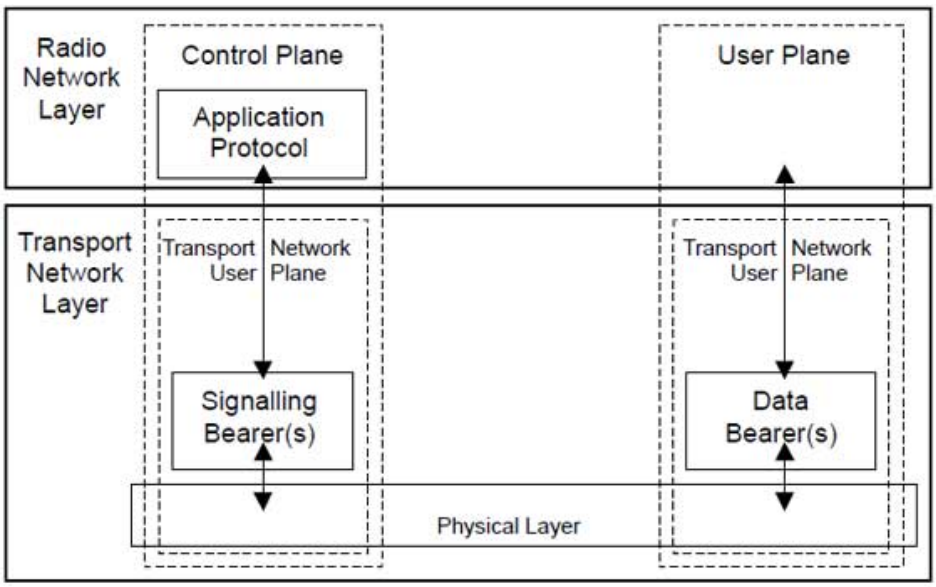



Figura 4.5: División de la arquitectura de protocolo radio [27].

Además, explicaremos la estructura de protocolos utilizada en E-UTRAN para soportar las interfaces S1 y X2. Se establece una separación entre la capa de red radio

(o *Radio Network Layer* (RNL)) y la capa de red de transporte (o *Transport Network Layer* (TNL)), tal y como se puede apreciar en la figura 4.6. Esta descomposición tiene como objetivo aislar las funciones que son específicas del sistema de comunicaciones móviles como puede ser LTE o UMTS, de aquellas que dependen de la tecnología de transporte utilizada como puede ser IP o ATM. Así, los protocolos específicos de la red de acceso radio constituyen la capa RNL, mientras que la capa TNL alberga los protocolos utilizados para el transporte de la información de la capa RNL entre las entidades de la red.

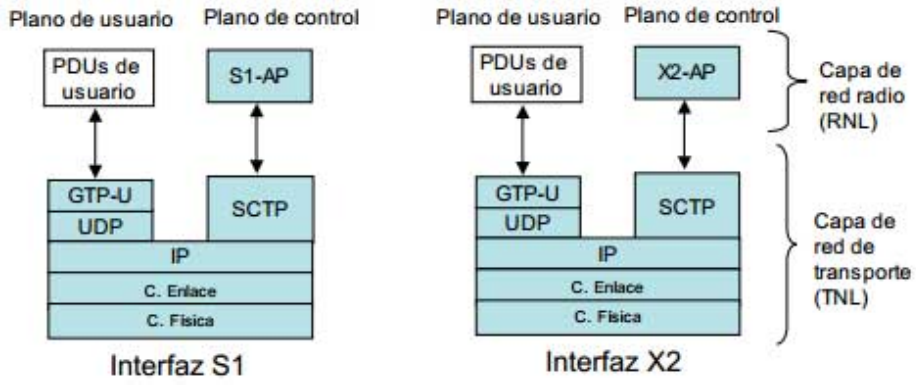



Figura 4.6: Protocolos en las interfaces S1 y X2 [4].

### 4.3.1. Plano de control

El plano de control es el encargado del envío de señalización NAS entre el equipo de usuario y el núcleo de red. Los protocolos NAS, se extienden entre el MME y el UE a través de la red troncal, los mensajes de éste protocolo se transportan de forma transparente en la interfaz radio, ya que son encapsulados dentro de la parte de datos de los mensajes RRC. Las principales funciones de estos protocolos son la autenticación, la autorización, la gestión de movilidad de los UE que no tienen una conexión RRC establecida, y la gestión de los servicios de la red LTE.

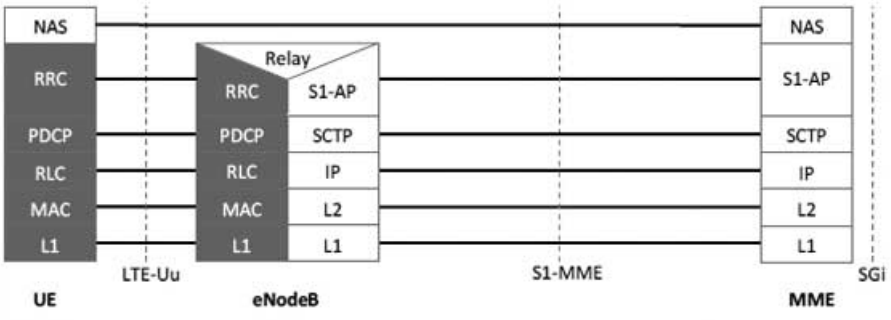



Figura 4.7: Pila de protocolos del plano de control [28].

Si nos situamos en la parte izquierda de la figura 4.7 observamos los protocolos sombreados, y son estos los que pertenecen a la red E-UTRAN, y concretamente a la interfaz radio. La interfaz que conecta los UE con los nodos eNB se denomina interfaz LTE-Uu.

### RRC

El protocolo NAS se encuentra en la cima de la pila de los procolos de la E-UTRAN, haciendo referencia a toda la señalización que se encuentra por encima de los eNB y que es encapsulada dentro de mensajes RRC en la interfaz radio. Los procotolos que se utilizan entre los equipos de usuario y los eNB se conocen como los protocolos del estrato de acceso o *Access Stratum* (AS), siendo la capa RRC la principal de este conjunto de protocolos ya que es la responsable de establecer *radio bearers* y configurar las capas inferiores.

De entre las funciones que podemos destacar de esta capa son las siguientes:

- Transmisión de información por canales *broadcast*.
- Paginación.
- Establecimiento, mantenimiento y liberación de una conexión RRC entre el UE y la E-UTRAN.
- Establecimiento, mantenimiento y liberación de *radio bearers*.
- Envío y control de informes de medidas del UE.
- Transmisión de mensajes NAS entre el UE y la red.
- Tratamiento de errores de los protocolos.
- Selección y reselección de la celda del UE y control del proceso.

En la capa RRC se ha simplificado el estado del equipo de usuario respecto al estándar UMTS a solamente dos, que aparecen en la tabla 4.1.

| Estado | Función |
|---|---|
| RRC_IDLE | El UE monitoriza el canal de paginación para detectar llamadas entrantes, adquirir información del sistema y realizar medidas de factores como la calidad del enlace de radio o el estado de las células vecinas para así poder realizar reselección de celda. |
| RRC_CONNECTED | La E-UTRAN es la encargada de la movilidad y no el UE. Éste último realiza medidas de calidad del canal, de las celdas vecinas y las transmite al eNB; además, monitoriza los canales de control asociados al canal de datos compartido. |

Tabla 4.1: Estados del UE en la capa RRC.

Los mensajes RRC se transmiten usando *Signaling Radio Bearers* (SRB), existiendo tres tipos distintos. El SRB0 se utiliza para mensajes que usen el canal común de control, como puede ser el establecimiento de una conexión RRC. El SRB1 se utiliza para transmitir mensajes RRC y mensajes NAS, hasta que se activa la seguridad; y finalmente el SRB2 se establece para transmitir mensajes NAS cuando la seguridad está activa, mientras que los RRC continuarán transmitiéndose por el SRB1.

Dependiendo de la información transmitida los elementos de información del sistema se agrupan en *Master Information Block* (MIB), y en diferentes bloques de información del sistema o *System Information Blocks* (SIB). Los MIB son los bloques más importantes y se transmiten cada 40 ms por el canal de *broadcast*. La información contenida

en ellos es esencial para que el UE pueda conectarse a la red; la transmisión se realiza en la trama con *System Frame Number* (SFN) 0 en la subtrama 0, y después en todas con SFN mod 4 = 0.

Existen 13 tipos de SIB:

- SIB1. Contiene parámetros para determinar si una celda es adecuada para el UE, y contiene información sobre la localización del resto de SIB, que están agrupados en mensajes *System Information* (SI). El SIB1 se transmite cada 80 ms en las tramas que SFN mod 8 = 0 en la subtrama 5.

- SIB2. Contiene información sobre los canales comunes y compartidos, se transmite en el primer mensaje SI como primera entrada, después del MIB.

- SIB3 al SIB8. Contienen información para la reselección de celdas.

- SIB9. Contiene el identificador del eNB.

- SIB10 - SIB12. Contienen notificacines e información de alertas y avisos.

- SIB13. COntienen información de los canales *multicast*.

#### PDCP

Los mensajes de la capa RRC son procesados por la capa *Packet Data Convergence Protocol* (PDCP), que proporciona funciones de cifrado, verificación, integridad de datos, compresión de cabeceras, funciones de soporte al *handover*, y descarte de datos del plano de usuario entre otras funcionalidades. Ésta se comunica con la capa *Radio Link Control* (RLC) a través de puntos de acceso al servicio o *Service Access Points* (SAP).

#### RLC

El *Service Access Points* (SAP) procesa los paquetes PDCP para que su tamaño sea el indicado por la capa MAC. La capa RLC configura la capa MAC y se comunica con ella a través de canales lógicos.

Las funciones de esta capa se realizan por entidades RLC, y disponen de tres modos de configuración:

- Modo transparente o *Transparent Mode* (TM). La capa RLC no realiza ninguna función, pero este modo únicamente está disponible para el plano de control.

- Modo sin reconocimiento o *Unacknowledged Mode* (UM). Proporciona un servicio de trasnferencia de datos unidireccional como puede ser un servicio punto a multipunto. Realiza segmentación y agrupación de paquetes PDCP, reordenación de paquetes RLC y detección de paquetes RLC duplicados.

- Modo reconocido o *Acknowledged Mode* (AM). Proporciona un servicio de transferencia de datos bidireccional, realiza las funciones del modo UM, en ambos sentidos. Además, incluye retransmisión de paquetes RLC, sondeos y generación de informes sobre el estado de recepción de paquetes.

#### MAC

Como se ha mencionado anteriormente, la capa MAC se encuentra debajo de la capa RLC y encima de la capa física. La capa RLC y la capa MAC se comunican mediante canales lógicos. Finalmente los paquetes generados por la capa MAC se preparan para enviarse por los canales de transporte hacia la capa física. Estos paquetes se conocen como bloques de transporte o *Transport Blocks* (TB). Esta capa se explicará con más detalle en los siguientes puntos.

En la parte derecha de la figura 4.7 nos encontramos con la interfaz *S1 Mobility Management Entity* (S1-MME) que se encarga de comunicar los eNB con la entidad MME, transmitiendo información de control. En este caso, ambas entidades se comunican a través de la capa *S1 Application Protocol* (S1-AP) que proporciona funcionalidades como el control de handover, búsqueda, etcétera. La transferencia de estos mensajes se realiza usando el protocolo de transporte *Stream Control Transmission Protocol* (SCTP), que ofrece un servicio de transferencia fiable al igual que *Transport Control Protocol* (TCP); aunque ofrece mejoras al proporcionar mayor robustez y versatilidad en distintos tipos de información. Esta capa proporciona mecanismos de control de flujo y de congestión, y permite multihoming y multistreaming.

La capa inferior a SCTP es la capa IP, que nos proporciona encaminamiento de la información del plano de control. Y en las capas inferiores nos encontramos con la capa MAC y la capa física, cuya comunicación ya hemos explicado.

### 4.3.2. Plano de usuario

La otra arquitectura en la que se divide la E-UTRAN conecta el eNB con las entidades S-GW y PDN-GW, para poder transmitir información desde y hacia el equipo terminal del usuario.

En la parte que pertenece a la red E-UTRAN, la pila de protocolos se aprecia en sombreado en la figura 4.8, ignorándose los protocolos que están por encima del protocolo IP ya que éste se encarga de enrutar el paquete a través de diferentes redes como puede ser Internet. La capa inmediatamente inferior, PDCP, procesa los paquetes IP realizando la compresión y descompresión de cabeceras, entre otras funcionalidades como se ha visto anteriormente. Además, esta capa realiza un descarte de datos si se ha superado un umbral en el tiempo de espera y un reordenamiento de *Packet Data Unit*s (PDUs), proporcionados por la capa RLC para su entrega a la capa IP. Por debajo de esta capa se encuentran la capa MAC y la capa física, cuyo funcionamiento es el mismo que en el plano de control. La interfaz que conecta el UE con el eNB se denomina interfaz Uu.

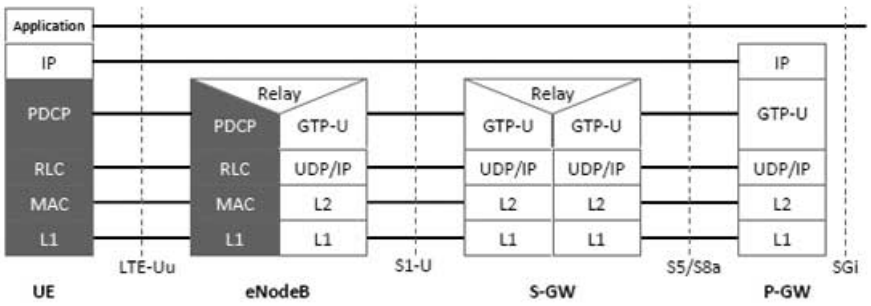



Figura 4.8: Pila de protocolos del plano de usuario.

Conforme observamos la figura 4.8 hacia la derecha nos encontramos con la interfaz S1-U que conecta el eNB con la entidad S-GW, que pertenece al núcleo de red ó EPC. La capa más alta que encontramos en la comunicación de estas entidades es la capa *GPRS Tunnelling Protocol - User Plane* (GTP-U), que es la encargada de proporcionar el encapsulado para el envío de paquetes IP de usuario. Este protocolo es heredado de GPRS, que en las redes GSM y UMTS se utiliza dentro del dominio de paquetes de la red troncal. Ésta capa se transporta sobre *User Datagram Protocol* (UDP)/IP y se utiliza para multiplexar los paquetes IP de múltiples usuarios. Al utilizar UDP no se utilizan mecanismos de entrega garantizada de paquetes, ni mecanismos de control de errores o control de flujo.

La interconexión entre las entidades S-GW y PDN-GW se realiza mediante la interfaz S5 cuando ambas pertenecen al mismo operador, y mediante la interfaz S8 cuando estas entidades se encuentran en redes distintas y existe un servicio de itinerancia.

En secciones anteriores hemos visto que la conexión con redes externas o redes IMS se realiza a través de la entidad PDN-GW mediante la interfaz SGi.

### 4.3.3. Interfaz S1

La interfaz S1 es la que conecta los eNB con la entidad MME. Como hemos visto en los apartados anteriores dicha interfaz se divide en dos partes:

- S1-U: Utilizada para la transmisión de datos por parte del usuario, en el plano de usuario.
- S1-MME: Utilizada para la transmisión de señalización en el plano de control.

Los protocolos que utilizan ambas partes se han visto anteriormente en el plano de usuario y plano de control, por lo que sabemos que el protocolo utilizado en una capa superior es IP, en ambos casos. Si continuamos con la pila de protocolos en el plano de usuario el protocolo que utiliza para el transporte es UDP, que da servicio al protocolo GTP-U. Éste protocolo está definido en por el 3GPP, y se utiliza también en la red *UMTS Terrestrial Radio Access Network* (UTRAN) facilitando así la interacción con dichas redes. Por otro lado, en el plano de control el protocolo inmediatamente superior a IP es SCTP, diseñado a partir de TCP pero con mayor robustez, seguridad, y fiabilidad para poder transportar los mensajes de señalización. El protocolo SCTP negocia previamente la identificación de los túneles que crea el protocolo GTP-U, el cual transmite los datos de los usuarios multiplexando la información de ellos. Por encima de SCTP encontramos la capa S1-AP.

#### S1-AP

Es un protocolo utilizado para dar servicios de señalización entre un eNB y el MME. Ofrece una serie de servicios y se pueden dividir según estén asociados a la señalización de un UE o no. Los principales servicios que ofrece este protocolo son:

- Gestión de *radio bearers*.
- Transferencia inicial de contextos.
- Indicación de información de capacidades del UE.
- Paginación.
- *Handovers*.
- Gestión de la interfaz S1.
- Balanceo de carga.
- Modificación del contexto del UE.
- Transferencia de estado.
- Traspaso de informes de localización.
- Envío de mensajes de alerta.

### 4.3.4. Interfaz X2

La interfaz X2 permite conectar los eNB entre sí. Esto permite que haya una baja latencia en *handovers*. Los protocolos utilizados en esta interfaz son mismo que en el interfaz S1 salvo que utiliza la interfaz *X2 Application Protocol* (X2-AP) en lugar de la interfaz S1-AP.

#### X2-AP

Entre las funciones más importanes de esta interfaz están la gestión de los *handovers* y la cooperación entre eNB vecinos. Se pretende reducir al máximo las interferencias entre celdas, para así poder utilizar el máximo espectro posible reservado para la red. Entre los principales servicios de este protocolo encontramos:

- Gestión de la movilidad.
- Gestión de la carga.
- Configuración y reinicio de la interfaz X2.
- Actualización de la configuración del eNB.

## 4.4. Capa MAC y capa física

La innovación de LTE es que está basada en *todo IP*, utilizando la conmutación de paquetes para todos los servicios. Esto implica que no se utilizan canales dedicados a cada usuario, sino que se comparte un solo canal físico entre múltiples usuarios. Dicha característica proporciona una mayor eficiencia en el uso de los recursos disponibles, y flexibilidad en la asignación del ancho de banda. La capa MAC se encuentra debajo de la capa RLC y encima de la capa física. Ésta se comunica mediante canales de transporte a su capa inferior. La descripción de la capa física de LTE en el estándar diferencia entre el duplexado por división en frecuencia o FDD, y el duplexado por división en tiempo o *Time Division Duplexing* (TDD). Nosotros explicaremos el duplexado por división en frecuencia ya que es el que hemos utilizado para el desarrollo del presente proyecto.

### 4.4.1. Tipos de canales

En LTE, se distinguen los canales según la información que contengan, la forma de procesar la información, y en función de en qué capa nos encontremos. Existen los siguientes tipos de canales:

- Canales lógicos. Conectan la capa RLC con la capa MAC, definen que tipo de información se transmite, son canales de tráfico, de control, de difusión, etcétera.
- Canales de transporte. Conectan la capa MAC con la capa física. En ellos se define como se transmite, podemos ver parámetros como la codificación, el entrelazado, etcétera.
- Canales físicos. En ellos se produce la transmisión de las señales de control y de datos, hacia el UE.

La relación entre los distintos canales y el mapeo entre cada una de las capas lo podemos observar en la figura 4.9.

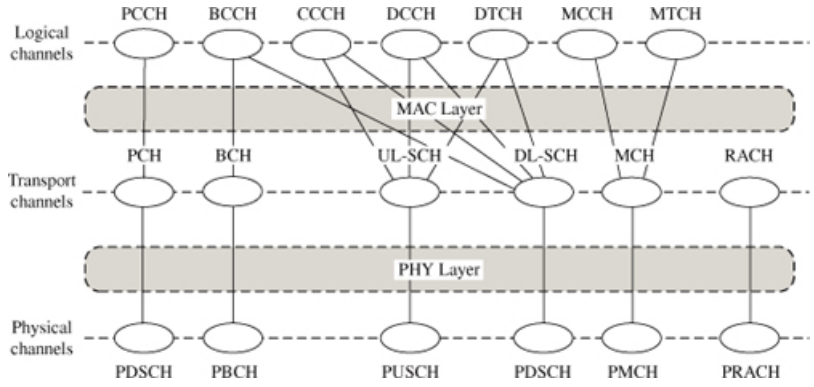



Figura 4.9: Mapeo de los distintos canales.

#### Canales lógicos

Los canales lógicos se pueden dividir según la información que transporten, y pueden dividirse en canales lógicos de control y canales lógicos de tráfico. La división de ellos la vemos en la tabla 4.2 , así como en el sentido en el que se utilizan.

| Nombre | Acrónimo | De control | De tráfico | DL | UL |
|---|---|---|---|---|---|
| *Broadcast Control Channel* | BCCH | x | | x | |
| *Paging Control Channel* | PCCH | x | | x | |
| *Common Control Channel* | CCCH | x | | x | x |
| *Multicast Control Channel* | MCCH | x | | x | |
| *Dedicated Control Channel* | DCCH | x | | x | x |
| *Dedicated Traffic Channel* | DTCH | | x | x | x |
| *Multicast Traffic Channel* | MTCH | | x | x | |

Tabla 4.2: Canales lógicos.

La utilización de los canales lógicos es la siguiente:

- BCCH (*Broadcast Control Channel*). Transmite información general sobre la red y la célula en el enlace descendente.
- PCCH (*Paging Control Channel*). Transmite información de aviso para los UE en el enlace descendente.
- CCCH (*Common Control Channel*). Se utiliza para el intercambio de información de control con los UE sin conexión RRC o que acceden por primera vez a una célula.
- MCCH (*Multicast Control Channel*).
- DTCH (*Dedicated Traffic Channel*). Transmite información con un UE determinado, es un canal dedicado tanto en el enlace descendente como en el ascendente.
- MTCH (*Multicast Traffic Channel*). Transmite información *multicast* en el enlace descendente.

### Canales de transporte

La capa MAC se comunica a través de los canales de transporte con la capa física. Éstos multiplexan la información según vayan a transmitirse. A continuación, en la tabla 4.3 se muestra la clasificación de dichos canales según en el enlace que se utilicen.

| Nombre | Acrónimo | DL | UL |
|---|---|---|---|
| *Broadcast Channel* | BCH | x | |
| *Dowlink Shared Channel* | DL-SCH | x | |
| *Paging Channel* | PCH | x | |
| *Multicast Channel* | MCH | x | |
| *Uplink Shared Channel* | UL-SCH | | x |
| *Random Access Channel* | RACH | | x |

Tabla 4.3: Canales de transporte.

### Canales físicos

En LTE la capa física se debe amoldar a la necesidad de transmisión de cada usuario, ya que no existen canales dedicados, con lo que debe proporcionar recursos físicos de forma adecuada a los canales de transporte. Por esta razón se especifican varios canales

en el sentido DL y en el UL. Se definen canales de transporte que tienen canales físicos específicos como lo son el *Broadcast Channel* (BCH) y el *Multicast Channel* (MCH). Como se puede apreciar en la figura 4.9, tienen los canales físicos *Physical Broadcast Channel* (PBCH) y el *Physical Multicast Channel* (PMCH) respectivamente. Los canales de transporte *Paging Channel* (PCH) y *Dowlink Shared Channel* (DL-SCH) comparten el canal físico *Physical Downlink Shared Channel* (PDSCH). Los canales de control en el enlace DL que no dan servicio a las capas superiores son *Physical Control Format Indicator Channel* (PCFICH), el *Physical Downlink Control Channel* (PDCCH) y el *Physical Hybrid ARQ Indicator Channel* (PHICH). Todas estas correspondencias se pueden apreciar en la tabla 4.4.

| **Nombre** | **Acrónimo** | **Tráfico** | **Control** | **DL** | **UL** |
|---|---|---|---|---|---|
| *Physical Broadcast Channel* | PBCH | x | | x | |
| *Physical Downlink Shared Channel* | PDSCH | x | | x | |
| *Physical Multicast Channel* | PMCH | x | | x | |
| *Physical Control Format Indicator Channel* | PCFICH | | x | x | |
| *Physical Downlink Control Channel* | PDCCH | | x | x | |
| *Physical Hybrid ARQ Indicator Channel* | PHICH | | x | x | |
| *Physical Uplink Shared Channel* | PUSCH | x | | | x |
| *Physical Uplink Control Channel* | PUCCH | | x | | x |
| *Physical Random Access Channel* | PRACH | | x | | x |

Tabla 4.4: Canales físicos.

Las descripciones y las principales funcionalidad de cada uno de los canales físicos mencionados en la tabla 4.4 son:

- PBCH (*Physical Broadcast Channel*). Transmite información básica de la red, como la canalización utilizada en una celda específica, la estructura del canal *Physical Hybrid ARQ Indicator Channel* (PHICH) y el número de identificación de trama.
- PDSCH (*Physical Downlink Shared Channel*). Transmite información de usuario entregada por la capa MAC mediante el canal de transporte *Dowlink Shared Channel* (DL-SCH), aunque puede transportar información de aviso. No es un canal dedicado y solamente se asigna al usuario cuando éste tiene que recibir algo.
- PMCH (*Physical Multicast Channel*). Transporta información de los servicios de difusión y multidifusión MBMS.
- PCFICH (*Physical Control Format Indicator Channel*). Transmite información al usuario sobre el número de símbolos utilizados para transmitir el canal *Physical Downlink Control Channel* (PDCCH).
- PDCCH (*Physical Downlink Control Channel*). Transmite información de control para el enlace descendente, como la asignación de recursos para los canales de aviso *Paging Channel* (PCH) y DL-SCH. Transmite la información *Acknowledgement* (ACK)/*Negative Acknowledgement* (NACK) para así, implementar el mecanismo *Hybrid Automatic Repeat Request* (HARQ) en el enlace de bajada.
- PHICH (*Physical Hybrid ARQ Indicator Channel*). Transporta los reconocimientos ACK/NACK para implementar el mecanismo HARQ en el enlace de subida.

- PUSCH (*Physical Uplink Shared Channel*). Este canal transmite información de usuario en el enlace de subida.
- PUCCH (*Physical Uplink Control Channel*). Transmite información de control en el enlace ascendente; como son las peticiones de asignación de recursos, reconocimientos (ACK/NACK), información de calidad del canal, etcétera.
- PRACH (*Physical Random Access Channel*). Transmite información del control en el enlace ascendente para el acceso inicial de los UE al eNB, para la ejecución de *handovers*, sincronización de transmisiones, etcétera.

### 4.4.2. Formación de trama

LTE se basa en la utilización de técnicas de acceso múltiple *Orthogonal Frequency Division Multiple Access* (OFDMA) en el enlace DL y *Single Carrier - Frequency Division Multiple Access* (SC-FDMA) en el enlace UL. Se especifica una separación entre subportadoras de $\Delta f = 15$ kHz, con una duración de símbolo de 66,67 $\mu$s que es la duración de un símbolo *Orthogonal Frequency Division Multiple* (OFDM), y un intervalo de guarda de 5 $\mu$s. Además, el número de portadoras a utilizar varía en función del ancho de banda del canal, pudiéndose acomodar por ello servicios con diferentes necesidades de ancho de banda. La anchura del canal varia según la asignación de recursos a las distintas transmisiones. En la tabla 4.5 se tiene la relación existente entre el BW, el número de puntos de la *Fast Fourier Transform* (FFT) (nº de muestras para la FFT) y el número de subportadoras.

| **BW** | **1,4 MHz** | **3 MHz** | **5 MHz** | **10 MHz** | **15 MHz** | **20 MHz** |
|---|---|---|---|---|---|---|
| **Nº ptos FFT** | 128 | 256 | 512 | 1024 | 1536 | 2048 |
| **Nº subportadoras** | 73 | 181 | 301 | 601 | 901 | 1201 |

Tabla 4.5: Relación BW-FFT-Subportadoras

En el dominio temporal, la señal se divide en tramas de 10 ms, que se subdividen en subtramas de 1 ms, un total de 10 subtramas. Éstas a su vez se dividen en 2 *slots* de 0,5 ms cada uno, por lo que una trama está compuesta por 20 intervalos de tiempo. Como se ha mencionado anteriormente, la duración de un símbolo OFDM más el intervalo de guarda se sitúa en torno a los 72 $\mu$s. Teniendo en cuenta que la separación entre subportadoras es de 15 kHz, se especifícan tres tipos de *slots*:

- De 7 símbolos utilizando un prefijo cíclico normal.
- De 6 símbolos utilizando un prefijo cíclico extendido.
- De 3 símbolos utilizando un prefijo cíclico super-extendido para las transmisiones multidifusión.

Debido a que la duración de cada símbolo es larga, se introduce un intervalo de guarda entre los mismos, que soluciona el problema de la interferencia inter-símbolo (ISI) al impedir que la cola de un símbolo se solape con el próximo. Además, se reducen los problemas de sincronización temporal. Durante este periodo de guarda, se puede transmitir el prefijo cíclico, que consiste en el final del símbolo OFDM copiado en dicho intervalo, y éste se transmite, seguido del símbolo OFDM.

Esta operación se realiza ya que en el receptor se integrará sobre un número entero de ciclos sinusoidales para cada camino de los multi-trayectos cuando se realiza la demodulación de OFDM con la FFT. Respetando la convolución cíclica, y manteniendo así la ortogonalidad de las portadoras.

En la figura 4.10 podemos ver la formación de una trama LTE, que como se ha mencionado anteriormente su duración es de 10ms, dividida en 10 subtramas, y 20 *slots*.

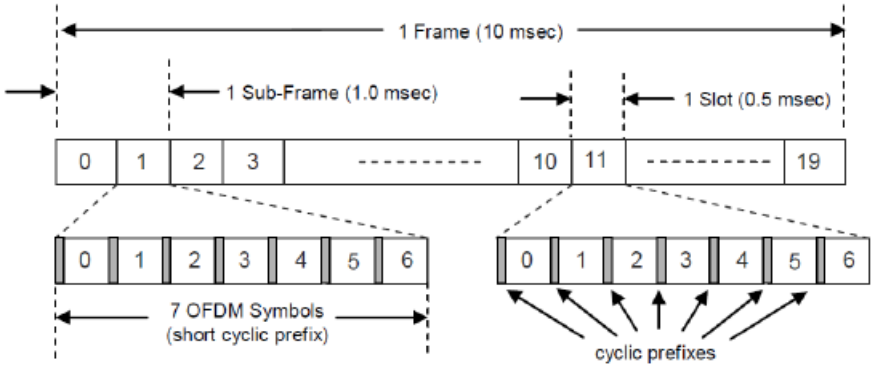



Figura 4.10: Estructura de la trama de LTE. 4.11

Las subportadoras se agrupan en conjuntos de 12 que suman un total de 180 kHz, durante un *slot* (0,5 ms). Esta agrupación se denomina *Resource Block* (RB). En función del ancho de banda disponible se transmitirán un determinado número de *Resource Block* (RB). Éstos bloques de recursos se componen de unidades más pequeñas, los *Resource Element* (RE), que es la asignación de un símbolo OFDM en el dominio del tiempo a una subportadora en el domino de la frecuencia. Esta relación en el tiempo y en la frecuencia forma una cuadrícula donde el elemento más pequeño es un *Resource Element* (RE). Hay que tener en cuenta que no todos son utilizados para enviar datos, sino que algunos se utilizan para control o enviar símbolos piloto. Esto se puede apreciar en la figura 4.11.

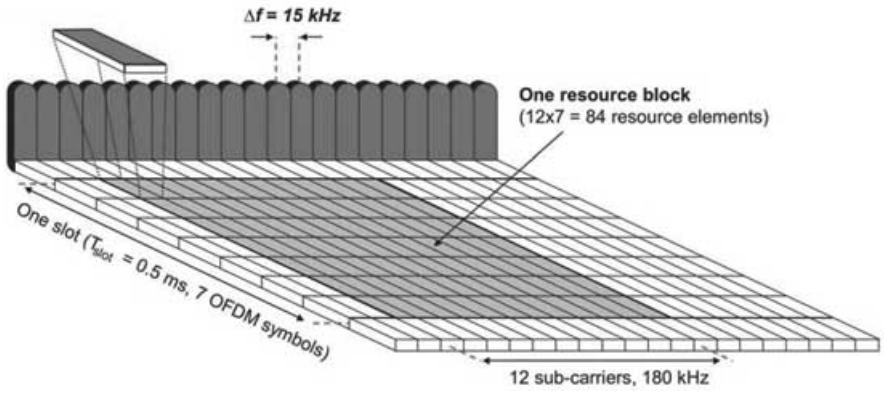



Figura 4.11: Estructura de recursos tiempo-frecuencia. 4.11

El caso más habitual es en el que se envían 7 símbolos por *slot*, por lo que las dimensiones de un RE son $15kHz \cdot \frac{0,5}{7}ms$. En consecuencia, un RB consta de $12 \cdot 7 = 84RE$. Los parámetros básicos de la modulación OFDM para LTE son:

- Número de bloques de recursos: $N_{RB}$
- Anchura de banda ocupada: $0,18 \cdot N_{RB}$ (MHz)
- Número de subportadoras: $N_{SC} = 12 \cdot N_{RB}$

La anchura de banda ocupada es menor que la anchura de banda normalizada para así dejar espacio a los límites del espectro. La tabla 4.6 muestra los valores de los parámetros anteriores para los diferentes anchos de banda que se pueden utilizar en LTE.

| **Ancho de banda nominal (MHz)** | 1,4 | 3 | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|
| **Ancho de banda ocupado (MHz)** | 1,08 | 2,7 | 4,5 | 9 | 13,5 | 18 |
| **Número de RB** | 6 | 15 | 25 | 50 | 75 | 100 |
| **Número de subportadoras** | 72 | 180 | 300 | 600 | 900 | 1200 |

Tabla 4.6: Parámetros de los diferentes anchos de banda en LTE.

Las capacidades brutas de LTE, incluyendo señalización y datos, se muestran en la tabla 4.7. Se recogen las capacidades teóricas máximas de LTE en Mbit/s con prefijo cíclico normal para las diferentes anchuras de banda normalizadas. Las tasas se han obtenido con la siguiente fórmula:

- $N_{RB}$ : número de RB.
- $N_{SP}$ : número de subportadoras que son por defecto 12.
- M : número de bits necesarios para representar la constelación (QPSK, 16-QAM o 64 QAM).
- 2 : número de slots por subtrama, que son dos (para obtener 1 ms).
- $N_{symb}^{TS}$ : número de símbolos por *slot*.

$$R = N_{RB} \cdot N_{SP} \cdot M \cdot 2 \cdot N_{symb}^{TS} \tag{4.1}$$

Como la subtrama dura 1ms, la tasa binaria (Kb/s) coincide con el número de bits.

| **BW (MHz)** | 1,4 | 3 | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|
| $N_{RB}$ | 6 | 15 | 25 | 50 | 75 | 100 |
| $R_{QPSK}$ | 2,016 | 5,040 | 8,400 | 16,800 | 25,200 | 33,600 |
| $R_{16QAM}$ | 4,032 | 10,080 | 16,800 | 33,600 | 50,400 | 67,200 |
| $R_{64QAM}$ | 6,048 | 15,120 | 25,200 | 50,400 | 75,600 | 100,800 |
| **MIMO 2x2** $R_{64QAM}$ | 12,096 | 30,240 | 50,400 | 100,800 | 151,200 | 201,600 |

Tabla 4.7: Tasas binarias máximas (Mb/s) con prefijo cíclico normal.

### 4.4.3. Señales de referencia y sincronismo

Las señales de referencia o símbolos piloto (RS) se utilizan en el enlace descendente para realizar la estimación del canal necesaria para la demodulación coherente. Estas señales siempre se modulan en QPSK. Existen tres tipos:

- Señales de referencia específicas de la celda. Son comunes a todos los usuarios de la celda, y contienen la identidad de la celda.

- Señales de referencia específicas de usuario.
- Señales específicas para el modo MBSFN.

Para poder acceder a una red LTE, es necesario realizar la búsqueda de celda, y se requiere poseer los sincronismos en tiempo y frecuencia para poder demodular el canal de difusión *Physical Broadcast Channel* (PBCH); obteniendo así la información de dicho sistema. En el enlace descendente se emiten dos señales de sincronismo además de la señal de referencia RS, y son:

- *Primary Synchronization Signal* (PSS) o señal de sincronismo primaria, la cual permite obtener la posición de la ventana de la FFT, el error de frecuencia, el sincronismo del *slot* y el identificador de celda dentro de un grupo de tres celdas. Se transmite dos veces por trama, en el último símbolo OFDM de los *slots* 0 y 10.
- *Secondary Synchronization Signal* (SSS) o señal de sincronismo secundaria. Ésta permite obtener el sincronismo de trama, el identificador del grupo de celdas, la longitud del prefijo cíclico, el modo de duplexado (TDD o FDD) y el sincronismo del PBCH. Se transmite dos veces por trama en el penúltimo símbolo OFDM de los *slots* 0 y 10.

Ambas señales de sincronismo se transmiten en el centro del radio canal, ocupando 62 subportadoras cada una de ellas, durante un símbolo OFDM, transmitiéndose cada 10 ms.

En el enlace ascendente hay dos:

- *Demodulation Reference Signal* (DMRS), se utiliza para la sincronización y estimación del canal UL.
- *Sounding Reference Signal* (SRS), que es empleada para la estimación del canal para la planificación de frecuencias.

### 4.4.4. Modulaciones en LTE

En LTE se utilizan modulaciones distintas cuando hablamos del enlace descendente o del ascendente. En el primero de ellos, se aplica la modulación OFDM mientras que en UL se utiliza la modulación SC-FDMA. A continuación, explicaremos las diferencias entre ellas y porqué se utilizan.

#### OFDM

La modulación OFDM tiene una serie de ventajas como son robustez frente a la propagación multicamino, facilidad de generación/demodulación mediante FFT, compatibilidad con técnicas MIMO y flexibilidad para adaptarse a radiocanales de gran ancho de banda gracias a la ecualización en el dominio de la frecuencia. Otra de sus ventajas es que nos permite desplegar redes isofrecuencia. La señal OFDM compleja tiene la siguiente expresión:

$$\widetilde{s}(t) = \sum_{l=-\infty}^{\infty} \left\{ \sum_{k=0}^{N-1} C_l^k \Phi_k(t)(t - lT) \right\} \tag{4.2}$$

Siendo $N$ el número de subportadoras, $T$ la duración del símbolo, y $C_l^k$ es el símbolo complejo de la modulación en el instante $l$ y subportadora $k$ y $\Phi_k(t) = rect_T(t) \cdot e^{j2\pi kt/T}$,

siendo $rect_T(t)$ un pulso unitario de duración $T$. Éstas dos últimas funciones son ortogonales. LTE puede utilizar las constelaciones: *Quadrature Phase Shift Keying* (QPSK), 16-*Quadrature Amplitude Modulation* (QAM) y 64-QAM. La señal OFDM es la suma de múltiples señales sinusoidales de diferentes frecuencias y amplitudes. Suponiendo que los símbolos de la modulación son incorrelados, se obtiene por superposición de los espectros de las subportadoras que son funciones de la forma (equivalente paso-bajo):

$$G_k(f) = \overline{|C_l^k|^2} \cdot T \cdot sinc^2(fT - k); (k = 0, 1, ..., N - 1) \tag{4.3}$$

Donde $\overline{|C_l^k|^2}$ es la potencia media de los símbolos de la modulación y $sinc(x) = sen(\pi x)/(\pi x)$. La separación entre subportadoras es $\Delta f = 1/T$, por lo que el ancho de banda total ocupado es $BW = N \cdot \Delta f$ más las colas del espectro de las subportadoras situadas en los extremos. Dado que cada subportadora transmite un flujo de bits independiente, los desvanecimientos en frecuencia afectan solamente a un subconjunto de bits codificados. Gracias a la redundancia del código FEC se puede recuperar la información. Además, es posible adaptar la modulación y la tasa del código FEC en función de la SNR en diferentes zonas del espectro maximizando así la capacidad del sistema.

La generación y demodulación de la señal OFDM se hace mediante el algoritmo FFT. Así, solamente en el intervalo de símbolo $lT \leq t < (l + 1)T$, la señal es:

$$\widetilde{s}(t) = \sum_{k=0}^{N-1} C_l^k \Phi_k(t - lT) = \sum_{k=0}^{N-1} C_l^k e^{j2\pi kt/T} \tag{4.4}$$

Si muestreamos la señal a $N$ muestras por símbolo, tenemos:

$$\widetilde{s}(lT + n\frac{T}{N}) = \sum_{k=0}^{N-1} C_l^k e^{j2\pi kn/N}; (0 \leq n \leq N - 1) \tag{4.5}$$

La expresión anterior es la IDFT de los símbolos complejos de la modulación y puede implementarse con la transformada de Fourier rápida inversa o (IFFT). A continuación, se introduce el prefijo cíclico debido al efecto de la propagación multicamino sobre la señal OFDM. En la demodulación se aplica el algoritmo DFT a las muestras de señal recibida, permitiendo así recuperar los símbolos transmitidos.

De esta forma se utiliza OFDMA en DL para conseguir que diversos UE usen simultáneamente una banda de frecuencias manteniendo la ortogonalidad entre cada una de sus señales. En el enlace ascendente o UL la señal puede sufrir distintos desplazamientos en frecuencia por efecto *Doppler* debido a los diferentes usuarios que realizan la transmisión hacia el eNB. Además, el sincronismo es más complejo debido a que es necesario sincronizar los bloques OFDM de los diferentes terminales que transmiten información al eNB.

### SC-FDMA

Por otra parte, el procesado de datos se lleva a cabo en la estación base debido a que se requiere de mayores recursos de computación y el procesado es más complejo. Con esto se consigue reducir el consumo de batería de los UE, además de simplificar el diseño de los dispositivos. Es por ello que en UL se ha implementado la modulación SC-FDMA, porque además de las ventajas anteriores mantiene la ortogonalidad entre transmisiones de diferentes usuarios, la posibilidad de ecualización en el dominio de la frecuencia, y la compatibilidad con técnicas MIMO.

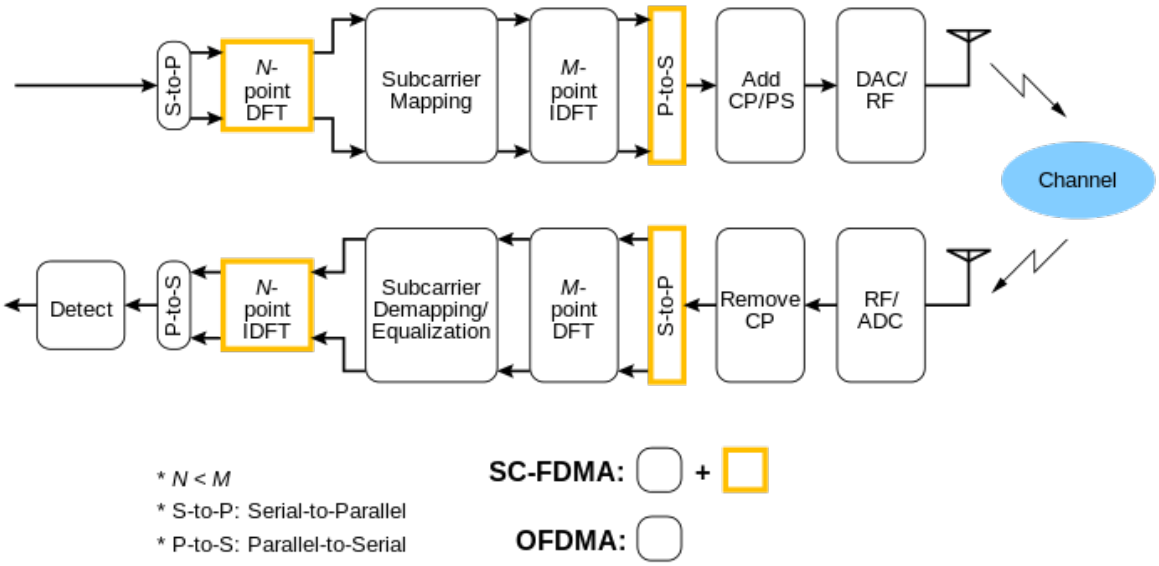



Figura 4.12: Procedimiento de SC-FDMA.

El proceso de transmisión de SC-FDMA es muy parecido al de OFDMA. La secuencia de bits transmitidos por un usuario se asigna a una constelación (BPSK o M-QAM), esta secuencia es transformada por un bloque de precodificación mediante un módulo de Transformada de Fourier Discreta o DFT. A continuación, la asignación de subportadoras se los diferentes valores, se puede realizar mediante un mapeo adyacente o un entrelazado. En el primer caso, se asignan un subconjunto de subportadoras consecutivas, mientras que en el segundo caso se asignan a subportadoras no continuas. A partir de aquí el proceso de transmisión es similar al de OFDMA: transformada inversa de Fourier (IDFT), y finalmente se añade un prefijo cíclico.

## 4.5. Funcionamiento del sistema LTE

### 4.5.1. Selección de celda

En primera instancia, el UE selecciona como prioridad su *Home Public Land Mobile Network* (HPLMN), que viene identificado por su *Mobile Country Code* (MCC) y *Mobile Network Code* (MNC) dentro de la USIM. Esta información se puede comparar con la difundida en las celdas, enviada en el bloque *System Information Blocks* (SIB)1, y dentro de la *Tracking Area* (TA) con identificador *Tracking Area Identifier* (TAI).

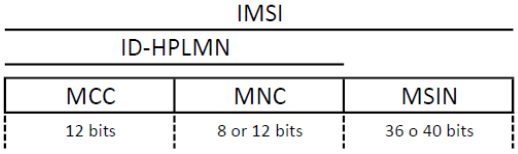



Figura 4.13: Estructura del IMSI.

El UE buscará las *Radio Access Technology* (RAT) que soporte las *Public Land Mobile Network* (PLMN), con lo que realiza un escaneo en todas las bandas soportadas y seleccionará en cada portadora la celda en la que reciba con mayor potencia. Una

vez seleccionado el PLMN el UE seleccionará la celda de mayor potencia para acceder a la red LTE.

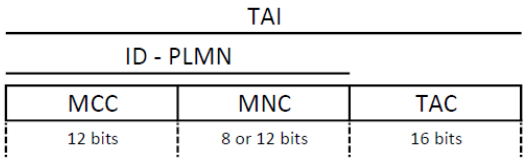



Figura 4.14: Estructura del TAI.

El UE escanea todos los canales de las bandas que soporta, y en cada portadora busca la celda de mayor potencia, o bien seleccionará la celda a partir de la información obtenida de las medidas anteriormente realizadas.

### 4.5.2. Registro/Desregistro en la red

Una vez seleccionada la celda satisfactoriamente el siguiente paso es realizar la solicitud de registro. El UE informa de su último *Globally Unique Temporary Identity* (GUTI) y, a partir de éste, el eNB conoce el último MME en el que estuvo registrado. Si el eNB está conectado a éste, le enviará la solicitud de registro. En caso contrario, seleccionará un nuevo MME e intentará conseguir el último contexto del UE. Si el UE proviene de UMTS/GPRS, en lugar del GUTI habrá indicado el *Packet Temporary Mobile Subscriber Identity* (P-TMSI). Si no es posible hallar un contexto válido se solicitará al UE su identificación mediante el IMSI.

A continuación, se realizarán las funciones de autenticación y cifrado enviando el MME el RAND y el AUTN hacia el UE para que le devuelva el resultado del desafío, i.e. el parámetro RES. Una vez finalizado, el MME seleccionará el S-GW más adecuado y le enviará la solicitud. Éste último la reenviará al P-GW, que le asignará una IP al UE. A partir de aquí ya se puede crear un contexto en el eNB e indicar la aceptación de registro al UE. El MME indicará el *Tunnel Endpoint Identifier* (TEID) del eNB al S-GW, y finalizará la creación del *default bearer*.

Por otra parte, el desregistro puede ser iniciado por el UE o por el MME; este procedimiento se compone de solicitud, eliminación del *default bearer* y confirmación del MME al UE.

### 4.5.3. Seguridad en LTE

En un sistema LTE existen varios mecanismos para implementar la seguridad:

- Cifrado. Se cifran tanto los datos del usuario como los mensajes de señalización, ya que existe información sensible perteneciente a un usuario específico.
- Integridad. Asegurar que la información de señalización recibida y/o enviada no está alterada, y que realmente la envió la entidad emisora. La integridad solo afecta a la señalización NAS y al protocolo RRC.
- Autenticación. Permite comprobar la identidad del usuario y asegurar que está conectado a la red autorizada.

### Autenticación

El proceso de autenticación, se puede ver en la figura 4.15. El detalle de los pasos seguidos se explica a continuación.

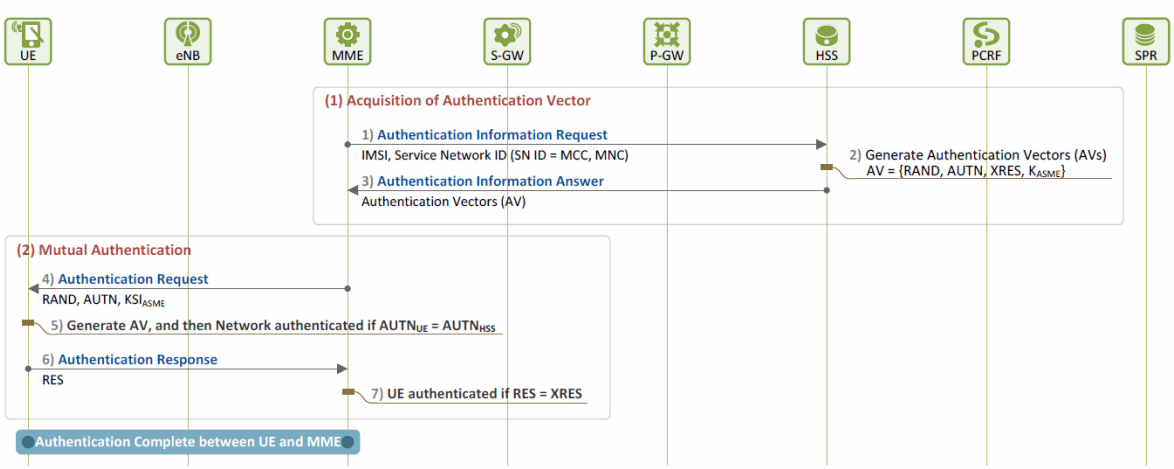



Figura 4.15: Procedimiento de autenticación.[23]

El MME es el encargado de iniciar el proceso siguiendo los siguientes pasos:

- Identificación del usuario. Se utiliza la información almacenada en el sistema (IMSI), o mediante el envío explicito de la información por parte del usuario.
- El MME solicita al HSS el vector de autenticación o *Authentication Vector* (AV) que contiene los siguientes elementos:
  - RAND, desafío aleatorio.
  - AUTN, prueba de autenticación que se le envía al usuario para que verifique la veracidad de la red.
  - XRES, resultado que deberá obtener el usuario cuando realice la autenticación con el AUTN.
  - CK, clave para el cifrado de la información. Se calcula a partir del RAND y una clave K que solo conoce el usuario y el HSS.
  - IK, clave de integridad utilizada para la comprobación de la integridad.
- El MME envía el desafío aleatorio RAND, junto con el AUTN al usuario.
- El usuario calcula el resultado de la autenticación (RES), utilizando el RAND y la clave K, y se lo envía al MME. También calcula CK y la clave IK.
- El MME compara RES y XRES, validando así al usuario.
- Finalmente el MME le asigna al usuario una identidad temporal denominada GUTI.

### Cifrado e integridad

Como hemos visto en el apartado anterior, utilizando K y RAND se calculan las claves CK e IK.

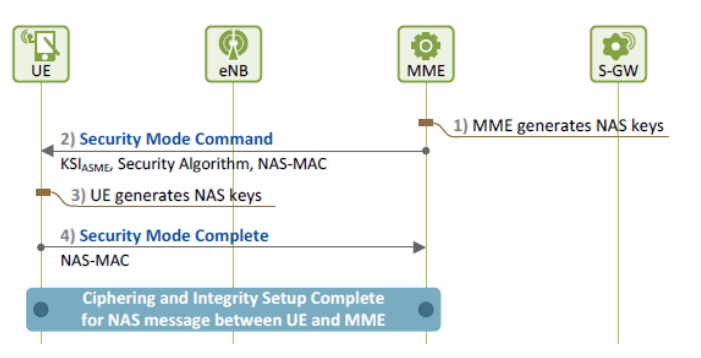



Figura 4.16: Procedimiento para el cifrado y la integridad.[23]

Utilizando estas dos últimas claves, el MME y el UE calculan una clave intermedia $K_{ASME}$. Utilizando ésta se calculan dos claves más:

- $K_{NASenc}$. Se utiliza para cifrar el tráfico NAS intercambiado entre usuario y MME.
- $K_{NASint}$. Se utiliza para proteger el tráfico NAS mediante un algoritmo de integridad.

Además, se calculan otras claves utilizadas para el eNB, que deriva otras dos claves de cifrado e integridad para información de control AS, y la clave de cifrado para el plano de usuario.

# Capítulo 5

# *Open Air Interface*

## 5.1. Introducción

En este capítulo se realiza un recorrido por los orígenes de la *Open Air Interface Software Alliance* (OSA), cómo surgió, quién la fundó, quienes la forman, y las distintas áreas estratégicas de las que se compone. Finalmente, explicaremos las partes en las que está dividido el *software* de OAI, y la multitud de plataformas en las que podemos utilizarlo.

## 5.2. La creación

EURECOM es una escuela de posgrado y un centro de investigación en sistemas de comunicaciones, ubicada en el parque científico internacional de Sophia Antipolis (Francia), y dentro del nuevo Campus SophiaTech. Fue fundada en 1991 como un consorcio y constituido por un amplio conjunto de socios[1] académicos y del ámbito industrial. Posteriormente debido a su embergadura EURECOM decidió centrarse en tres actividades: redes y seguridad, comunicaciones multimedia y comunicaciones móviles.


Figura 5.1: Logo EURECOM

[1]SFR, Orange, ST Microelectronics, BMW Group Research & Technology, Symantec, Monaco Telecom, SAP, IABG, Telecom ParisTech, Aalto University (Helsinki), Politecnico di Torino, Technische Universität München (TUM), Norwegian University of Science and Technology (NTNU), Vietnam National University Ho Chi Minh Vile (VNU), Chalmers University, Principality of Monaco, Institut Mines Telecom

El departamento de comunicaciones móviles es el mayor de todos, y está centrado en el procesamiento de sistemas celulares radio de cuarta generación (4G) y de la siguiente, la 5G. Además, se centra en protocolos inalámbricos, teoría de la información y redes. Su financiación para la investigación proviene de la industria privada y de fuentes nacionales y europeas.

La observación de la rápida evolución de los sistemas inalámbricos, y la demanda de soluciones abiertas y flexibles, llevó a EURECOM a crear, en el año 2014, la *Open Air Interface Software Alliance* como una entidad jurídicamente independiente y sin fines lucrativos. Además, decidió transferir gratuitamente todos los proyectos de 4G desarrollados bajo una licencia *software* de código libre.


Figura 5.2: Logo *Open Air Interface*.

La OSA actualmente proporciona un ecosistema de código libre para los protocolos del núcleo de red o EPC, y para la red de acceso o E-UTRAN de los sistemas celulares 3GPP, basados en los estándares de la Rel.10 para LTE. El *software* de código libre simplifica el acceso a la red, reduce los costes, aumenta la flexibilidad y mejora la velocidad de innovación cuando hablamos de introducir nuevos servicios.

## 5.3. ¿Qué es OAI?

Dado que la OSA está inmersa tanto en el mundo de la industria como en el académico, y la constante evolución de las redes móviles hacia 5G, existen diferentes áreas en las que los miembros de ésta crean nuevos proyectos. Las áreas estratégicas de la OSA se muestran en la figura 5.3.

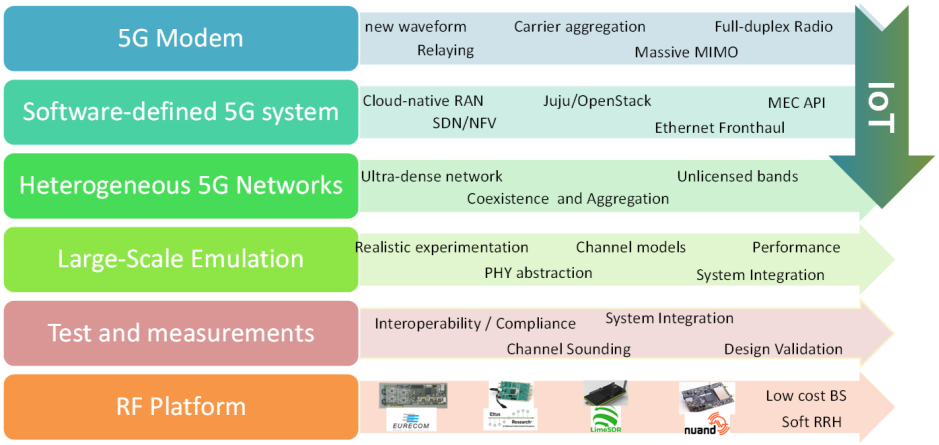



Figura 5.3: Áreas estratégicas de la *Open Air Interface Software Alliance*.

En nuestro caso, nos interesa el desarrollo del *software* libre basándose en el estándar 3GPP en las partes del UE, el eNB y del CN, ya que a partir de ahí hemos desarrollado el presente proyecto. La OSA proporciona un software para el EPC de código libre denominado *openairCN*, compatible con los equipos estándar MME, HSS, S-GW, P-GW de la Rel. 10. Este se distribuye bajo una licencia *Apache v2.0*, con el fin de facilitar la integración con un entorno *OpenStack* Por otra parte, el software para la red de acceso se encuentra bajo el nombre de *openair5G* y es compatible con los equipos UE y eNB de la Rel. 10. Se distribuye libremente por la OSA según los acuerdos de propiedad intelectual utilizados en la normativa del 3GPP.

## 5.4. Partes de OAI

El *software* que ofrece OAI está escrito en lenguaje C para Linux con un kernel de baja latencia o *low-latency kernel* y optimizado para procesadores Intel x86 y ARM. El código fuente está organizado en los siguientes directorios:

- **cmake-targets:** herramientas para la compilación, simulación, emulación y plataformas de tiempo real, generando dichos archivos para la compilación.
- **common:** código común de todas las capas.
- **openair1:** código de la capa física, *scheduling* y *PHY abstraction* para OAISIM.
- **openair2:** código de la capa RLC, PDCP, RRC, implementando el protocolo X2-AP.
- **openair3:** código *middleware* tanto para el eNB como para el UE, implementando los protocolos S1-AP, NAS, GTP-U v1.
- **openair-cn:** código fuente de protocolos del CN.
- **targets:** código específico para ejecutables, contiene configuraciones y compilaciones antiguas del sistema.

## 5.5. Distintas posibilidades

### Selección del escenario

El *software* de OAI nos ofrece un amplio abanico de posibilidades a la hora de montar nuestro escenario, proporcionando un alto grado de flexibilidad, permitiendo adaptarse a los recursos *hardware* de los que dispongamos. Las diferentes configuraciones en función de los componentes son:

- OAI EPC + OAI eNB + OAI UE
- OAI EPC + OAI eNB + UE (comercial)
- OAI EPC + eNB (comercial) + OAI UE
- OAI EPC + eNB (comercial) + UE (comercial)
- EPC (comercial) + OAI eNB + OAI UE
- EPC (comercial) + OAI eNB + UE (comercial)
- EPC (comercial) + eNB (comercial) + OAI UE

En el presente proyecto realizaremos el análisis de los dos primeros escenarios, ya que no disponemos en el laboratorio de eNBs y EPCs comerciales.

### Variedad de plataformas

OAI es una implementación de código libre que cubre las diferentes partes de la pila de protocolos del 3GPP como son los equipos eNB, UE, MME, HSS, S-GW, P-GW. El *software* funciona en distintas plataformas SDR (USRP, BladeRF, EXMIMO y LimeSDR), ejecutándose en una plataforma informática de uso general Linux con un núcleo de baja latencia. En la figura 5.4 podemos ver la variedad de plataformas SDR que soporta el *software* OAI.

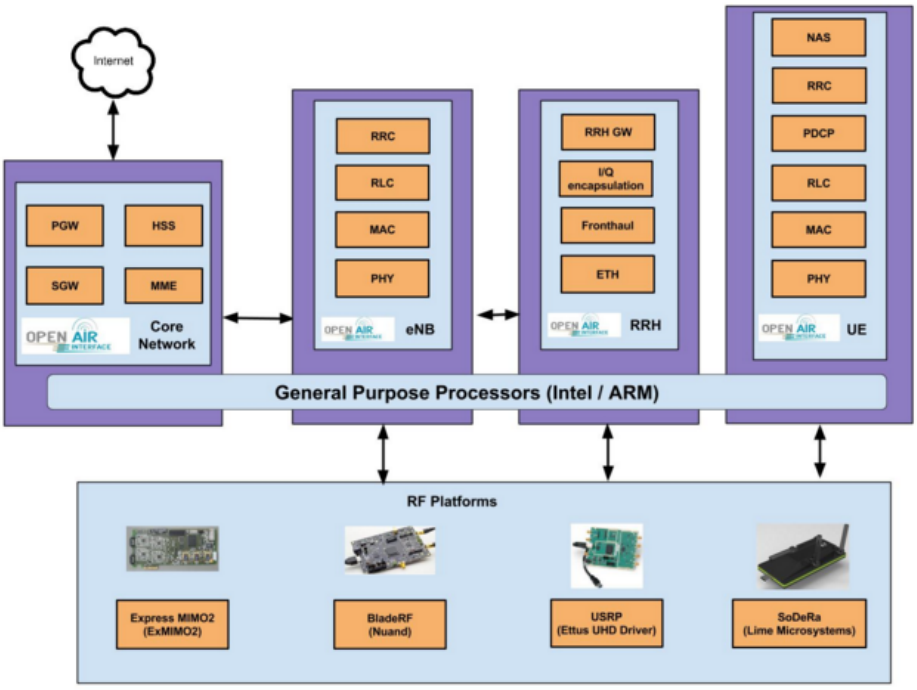



Figura 5.4: Plataformas SDR soportadas por OAI

# Capítulo 6

# Instalación y configuración de OAI

## 6.1. Introducción

En este capítulo, antes de comenzar con la instalación y configuración de OAI y OAISIM, se realiza la puesta a punto de las máquinas virtuales para que puedan ser utilizadas en la simulación de un entorno real. Acto seguido, explicaremos con detalle la instalación de OAI y OAISIM en las máquinas virtuales preparándolas así para el posterior análisis. Finalmente detallaremos la instalación y configuración del escenario real, en el que se configuraran el OAI-CN, OAI-eNB y los UE.

## 6.2. Máquinas virtuales

### Características técnicas del equipo real para simulación

El equipo que se va a utilizar para instalar las máquinas virtuales y posteriormente la creación y simulación del escenario real es un equipo portátil de la marca Toshiba, sus características se pueden consultar en la sección 3.3, subsección PC portátil *PC portátil - Toshiba L850*.

### Características técnicas de las máquinas virtuales

Para la creación de las máquinas virtuales se ha utilizado el *software* de virtualización gratuito para uso personal *VMware Workstation 12 Player*, ya que nos proporciona un interfaz de usuario simplificada y con independencia del sistema operativo instalado. Los recursos utilizados en cada una de las máquinas virtuales (MV) que crearemos para realizar la simulación son:

- **Núcleos utilizados:** 2
- **Memoria RAM:** 3 GB
- **Memoria HDD:** 30 GB
- **Tarjetas de red:** 2

### Puesta a punto de las máquinas virtuales

Una vez creadas las dos máquinas virtuales, procederemos con la instalación de *Ubuntu Linux 14.04. LTS (64 bits)*, ya que este sistema operativo se distribuye como *software libre*. Podemos descargarlo de la página oficial de Linux.

Posteriormente, procederemos a modificar la configuración de red de ambas máquinas, para obtener el diseño visto en la figura 6.1. La configuración en cada una de ellas es diferente, ya que el host *EPC* tiene una tarjeta de red *eth0* para la conexión a Internet, y otra tarjeta (*eth1*) para la red interna con la que tiene conexión con el host *OAISIM*. En cambio el host *OAISIM* dispone únicamente de una tarjeta de red para la red interna y con la que se comunica con el host *EPC*, la tarjeta *eth0*.

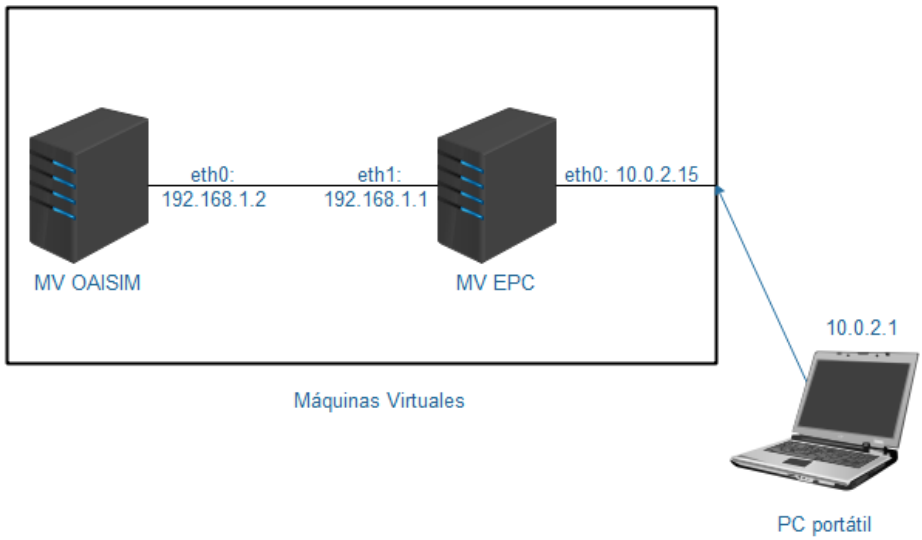



Figura 6.1: Estructura lógica de la conexión de las máquinas virtuales.

Para establecer una dirección IP fija en un sistema operativo *Linux*, podemos hacerlo de dos formas; una es mediante un terminal y escribir los comandos correspondientes, y otra es modificar el archivo de configuración `/etc/network/interfaces`.

En la máquina virtual *EPC*, y en el archivo anteriormente mencionado, debemos de introducir la información detallada en la tabla 6.1. El anexo A contiene la información de red para el archivo `/etc/network/interfaces`.

| **Tarjeta de red "eth0"** | |
|---|---|
| Tarjeta de red para la conexión a Internet | eth0 |
| Dirección IP estática | 10.0.2.15 |
| Máscara de red | /24 |
| Dirección IP de red | 10.0.2.0 |
| Dirección IP *broadcast* | 10.0.2.255 |
| **Tarjeta de red "eth1"** | |
| Tarjeta de red para la conexión a Internet | eth1 |
| Dirección IP estática | 192.168.1.1 |
| Máscara de red | /24 |
| Dirección IP de red | 192.168.1.0 |
| Dirección IP *broadcast* | 192.168.1.255 |

Tabla 6.1: Información de red de la MV EPC.

Una vez comprobado que las direcciones IP son correctas guardamos el archivo. Abrimos el terminal y reiniciamos los interfaces del equipo de la siguiente manera:

```
$ sudo ifconfig eth0 down
$ sudo ifconfig eth0 up
```

```
$ sudo ifconfig eth1 down
$ sudo ifconfig eth1 up
```

A continuación, comprobamos que las interfaces de red tienen la configuración correcta. En el mismo terminal escribimos:

```
$ ifconfig -a
```

Para terminar con la configuración de red de esta máquina añadimos la ruta por defecto, que en nuestro caso es:

```
$ sudo route add default gw 10.0.2.1
```

Ahora procederemos a modificar el archivo de configuración de red de la máquina virtual OAISIM; debemos incluir la información acogida en la tabla 6.2, de la misma forma que en el caso de la MV EPC.

| **Tarjeta de red "eth1"** | |
|---|---|
| Tarjeta de red para la conexión a Internet | eth1 |
| Dirección IP estática | 192.168.1.2 |
| Máscara de red | /24 |
| Dirección IP de red | 192.168.1.0 |
| Dirección IP *broadcast* | 192.168.1.255 |
| Puerta de enlace | 192.168.1.1 |

Tabla 6.2: Información de red de la MV OAISIM.

Y después añadimos la ruta por defecto, que en este caso es:

```
$ sudo route add default gw 192.168.1.1
```

De nuevo reiniciamos los interfaces de red; bien con los comandos anteriores o con el siguiente:

```
$ sudo /etc/init.d/networking restart
```

Una vez terminada la configuración de red en ambas máquinas comprobamos que tienen conectividad entre sí, y que la máquina EPC tiene conexión a Internet. Para ello nos bastaría con realizar una prueba de conexión con la herramienta *ping*.

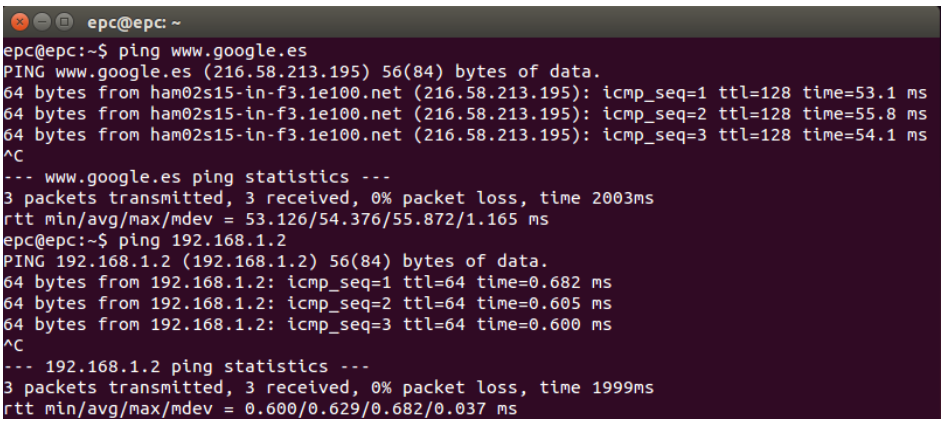



Figura 6.2: Test de ping en la máquina virtual EPC.

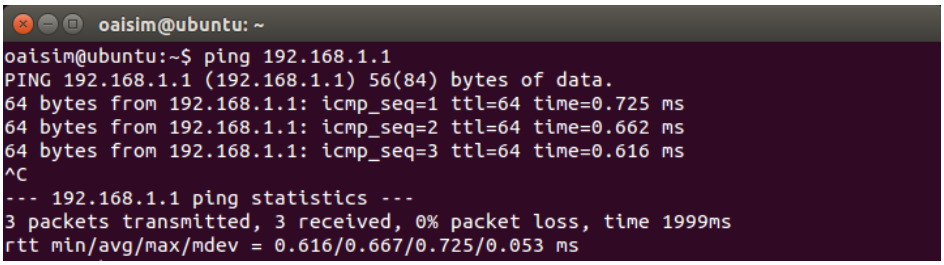



Figura 6.3: Test de ping en la máquina virtual OAISIM.

Por otro lado, antes de proceder con la instalación de OAI, debemos de realizar las siguientes tareas: instalación de un kernel de baja latencia, desactivación del escalado en frecuencia de la CPU, y de la desactivación de los estados C de la BIOS; estas tres premisas las realizaremos en las máquinas a utilizar.

### Instalación de kernel de baja latencia

El kernel que se recomienda para OAI con Ubuntu 14.04 LTS es el kernel 3.19 de baja latencia. También recomiendan para desarrolladores el sistema operativo Ubuntu 16.04 con el kernel 3.8. En nuestro caso hemos utilizado el primero. Para la instalación del kernel debemos escribir la siguiente línea de comandos:

```
$ sudo apt-get install linux-image-3.19.0-61-lowlatency linux-headers-3.19.0-61-
    lowlatency
```

A continuación reiniciamos nuestra máquina. Una vez hecho esto abrimos un terminal y comprobamos mediante el siguiente comando que la instalación del kernel ha sido positiva.

```
$ uname -a
```

### Desactivación de los estados C, gestíon de energía

En segundo lugar, vamos a proceder a la desactivación de los estados C de la BIOS, desactivando las funciones de administración de energía. Para ello abrimos el archivo

`etc/default/grub` y escribimos la siguiente línea en dicho fichero.

```
GRUB_LINUX_DEFAULT=‘‘quiet intel_pstate=disable processor.max_cstate=1 intel_idle.
    max_cstate=0 idle=poll"
```

Acto seguido, tecleamos en el terminal el comando:

```
$ update-grub
```

Después añadiremos al final del archivo `/etc/modprobe.d/backlist.conf` la siguiente línea (si no existe el fichero lo crearemos).

```
$ blacklist intel_powerclamp
```

### Desactivando el escalado en frecuencia

Para poder comprobar la información de la CPU, hemos instalado la herramienta *i7z*. Para ello escribimos en la terminal lo siguiente:

```
$ sudo apt-get install i7z
```

Para acceder a esta herramienta la invocamos con:

```
$ sudo i7z
```

Para desactivar el escalado en frecuencia hemos instalado la herramienta *cpufrequtils*, como sigue:

```
$ sudo apt-get install cpufrequtils
```

Seguidamente editaremos el archivo `/etc/default/cpufrequtils`, y añadiremos la siguiente línea:

```
GOVERNOR = "performance"
```

Una vez que hemos editado y guardado los cambios del archivo, necesitamos desactivar el demonio *"Ondemand"*; si no, una vez que hemos reiniciado el equipo los ajustes podrían sobrescribirse. Con el siguiente comando desactivamos el demonio:

```
$ sudo update-rc.d ondemand disable
```

Una vez realizado todos estos pasos reiniciamos el equipo, y comprobamos que todo se encuentra configurado correctamente. Para ello hemos utilizado los siguientes comandos:

```
$ cpufreq-info
$ sudo i7z
```

## 6.3. EPC + OAISIM

En primer lugar, hemos realizado la simulación del escenario real en máquinas virtuales. Hemos realizado la instalación de OAI CN y de OAISIM en máquinas virtuales diferentes, para posteriormente trasladar dicha configuración a los equipos reales situados en el laboratorio. El diseño de las máquinas virtuales dentro de nuestro equipo portátil se puede apreciar en la figura 6.4. El *Network Address Translation* (NAT) lo realiza automáticamente el programa *VMware* para permitir el acceso a Internet a las máquinas virtuales.

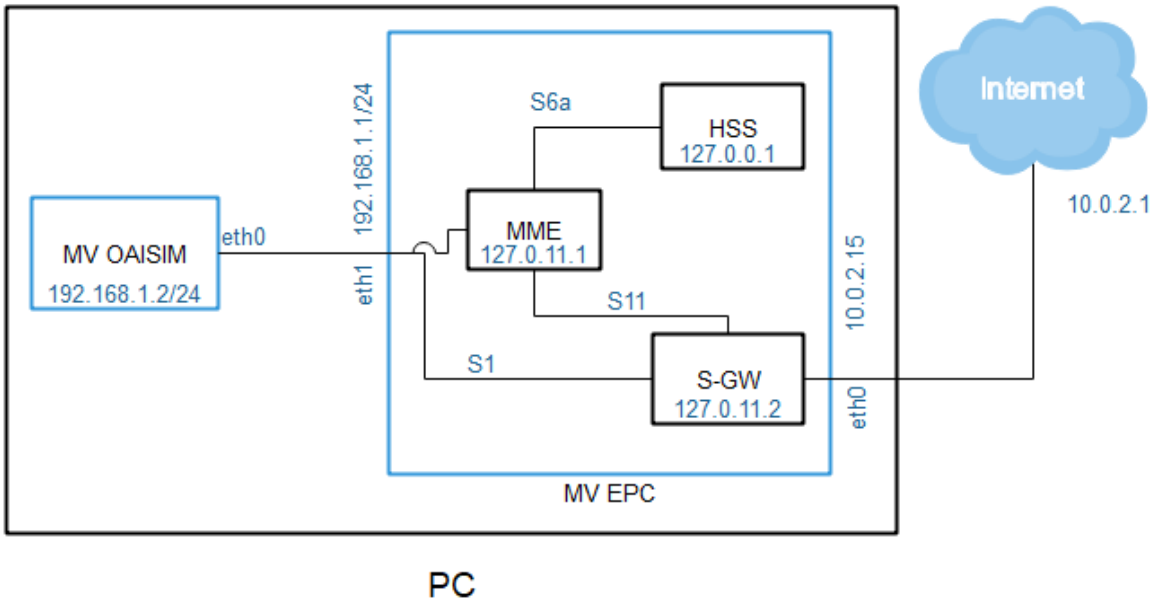



Figura 6.4: Esquema de red de las máquinas virtuales.

### 6.3.1. Instalación de OAI CN

En primer lugar, y antes de comenzar con la instalación del CN de OAI, instalaremos *git* para poder descargar el *software*. En el equipo llamado EPC, abrimos un terminal y escribimos el siguiente comando para dicha instalación.

```
$ sudo apt-get install git
```

En segundo lugar, y una vez instalado *git*, procedemos a descargar los códigos fuente con el siguiente comando:

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

Y para añadir herramientas opcionales a OAI escribimos:

```
$ git clone https://gitlab.eurecom.fr/oai/xtables-addons-oai.git
```

En tercer lugar, necesitamos un nombre de dominio para el EPC, llamado *Fully Qualified Domain Name* (FQDN). Para ello abrimos el archivo `/etc/hosts` en el equipo llamado *"epc"* y escribimos la siguiente configuración ajustándonos al plano de topología de la figura 6.4.

```
127.0.0.1 localhost
127.0.11.1 mme.5GLaboratory mme
127.0.11.2 hss.5GLaboratory hss
```

Reiniciamos el sistema para que los cambios tengan efecto. A continuación comprobamos que la configuración es correcta con el comando:

```
$ hostname -f
```

En nuestro caso el resultado obtenido es *epc.5GLaboratory*. Finalmente antes de realizar la instalación en nuestro caso hemos creado una carpeta en `/usr/local/etc/` llamada `oai`, en la cual se guardarán una serie de archivos posteriormente como los certificados de autenticación o los distintos archivos de configuración. Una vez creado esta carpeta para realizar la instalación de *openair-cn*, nos dirigiremos al directorio donde hemos descargado el repositorio y se encuentran los códigos fuente. Una vez situados en la carpeta `/openar-cn/SCRIPTS` ejecutaremos los siguientes scripts en un terminal.

```
$ ./build_mme -i
$ ./build_hss -i
$ ./build_spgw -i
```

La opción *"-i"* establecida en la ejecución de los tres scripts anteriores se utiliza únicamente la primera vez, ya que instala por primera vez una serie de paquetes necesarios para la compilación y posterior puesta en funcionamiento de los distintos equipos (MME, HSS, S-GW).

### 6.3.2. Instalación de OAISIM

En este lugar, realizamos la instalación de OAISIM en la segunda máquina virtual. Una vez realizado la desactivación de la gestión de energía de los estados C de la BIOS, de la desactivación del escalado de frecuencia y de la instalación del kernel de baja latencia, como se ha visto en puntos anteriores, procederemos con la descarga e instalación. Para descargar el código fuente, necesitamos tener instalado *git*, por lo que se procede de la misma manera. El comando para descargar e instalarlo es el siguiente:

```
$ sudo apt-get install git
```

En segundo lugar, descargaremos el código fuente que además de incluir los scripts de OAISIM incluye los archivos para compilar y arrancar el eNB y el UE. El comando para descargar el repositorio es:

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

En tercer lugar, una vez descargado el repositorio y situados en la carpeta de descargas, abrimos un terminal y nos situamos en el directorio `.../openairinterface5g/cmake_targets`. Y para proceder con la compilación de OAISIM y utilizando la interfaz S1 en nuestro caso (existe la posibilidad de no utilizar dicha interfaz) escribimos las siguientes líneas de comandos*:

```
$ ./build_oai -I
```

*Nota: Ejecutar solamente la primera vez para la instalación de paquetes

### 6.3.3. Configuración de OAI CN

#### 1. Configuración del HSS

En primer lugar, lo que debemos hacer para comenzar con la configuración del CN (en el equipo *"epc"*) es la modificación de la base de datos que utilizará el HSS. Para ello debemos de tener instalada la herramienta *mysql*, que nos ayudará con la modificación, creación, o eliminación de tablas que utilizaremos en nuestra base de datos. En nuestro caso hemos lo hemos instalado mediante el comando:

```
$ sudo apt-get install mysql-server mysql-common mysql-client
```

A continuación arrancamos el servidor de *mysql*.

```
$ sudo /etc/init.d/mysql restart
```

Una vez concluida la puesta en funcionamiento del servidor, procedemos rellenar las tablas. En la tabla *mmeidentity* incluiremos el nombre de los equipos involucrados y el dominio de éstos, lo vemos en la tabla 6.3.

| Equipo | Dominio |
|---|---|
| epc.5GLaboratory | 5GLaboratory |
| hss.5GLaboratory | 5GLaboratory |

Tabla 6.3: Equipos y dominios para la tabla *mmeidentity*.

Para incluir los equipos anteriores, arrancamos *mysql* con el *root* y *password* correspondientes que hayamos establecido durante la instalación. La base de datos que utiliza OAI por defecto se encuentra en la ruta `/openair-cn/SRC/OAI_HSS/db/oai_db.sql`.

Para poder importar la base de datos debemos escribir el siguiente comando:

```
$ mysql -u usuario -p nombre\_basededatos < data.sql
```

Donde "usuario" es el usuario de la base de datos, "nombre_basededatos" el nombre de la base de datos y "data.sql" es el nombre de la copia de la base de datos.

Una vez dentro de mysql, seleccionaremos la tabla nombrada anteriormente e introduciremos los valores con las siguientes líneas de comandos.

```
$ mysql > INSERT INTO mmeidentity (mmehost,mmerealm,UE-Reachability) VALUES  ('epc.5
    GLaboratory','5GLaboratory',0);
$ mysql > INSERT INTO mmeidentity (mmehost,mmerealm,UE-Reachability) VALUES ('hss.5
    GLaboratory','5GLaboratory',0);
```

Posteriormente, en la tabla *pdn* debemos de introducir el *International Mobile Subscriber Identity* (IMSI) al que permitiremos conectarse a nuestra estación base. Esta tabla la modificaremos en el escenario real, ya que en la simulación usaremos los datos por defecto; lo mismo que ocurre con la tabla *users*.

Una vez concluida la modificación e introducción de datos en la base de datos vamos a proceder a modificar el archivo de configuración del equipo virtual HSS, que se

encuentra en la ruta `/usr/local/etc/oai/hss.conf`. Los parámetros a modificar son los siguientes:

- MYSQL_server → Dirección IP del servidor *mysql*.
- MYSQL_user → Nombre de usuario de la base de datos.
- MYSQL_pass → Clave de usuario de la base de datos.
- MYSQL_db → Nombre de la base de datos.

Los valores en nuestro caso se muestran a continuación:

- MYSQL_server = "127.0.0.1";
- MYSQL_user = "root";
- MYSQL_pass = "5GLaboratory";
- MYSQL_db = "oai_db";

Otro archivo a modificar antes de poner en funcionamiento el HSS es el `hss_fd.conf`, que en nuestro caso se encuentra ubicado en `/usr/local/etc/oai`. Los parámetros que hemos modificado son:

- Identity → Nombre del equipo HSS válida en elFQDN.
- Realm → Nombre de dominio.
- TLS Cred → Ruta de los archivos que contienen la clave y certificado *Transport Layer Securtiy* (TLS).
- TLS CA → Ruta del certificado perteneciente a la autoridad certificadora para TLS.
- Port → Puerto de escucha para las conexiones.
- ConnectPeer → Nombre del equipo MME al que nos conectaremos.
- ConnectTo → Dirección IP del MME.

Los valores que hemos dado a los parámetros anteriormente mencionados son los siguientes:

- Identity = "hss.5GLaboratory";
- Realm = "5GLaboratory";
- TLS Cred = "/usr/local/etc/oai/freeDiameter/hss.cert.pem", "/usr/local/etc/oai/freeDiameter/hss.key.pem";
- TLS CA = "/usr/local/etc/oai/freeDiameter7hss.cacert.pem";
- Port = "3868";
- ConnectPeer = "epc.5GLaboratory";
- ConnectTo = "127.0.0.1";

La posterior acción a realizar una vez concluida la modificación de los archivos anteriores es crear las claves y certificados de autenticación. En la terminal nos situaremos en la carpeta `.../openair-cn/SCRIPTS` y ejecutaremos el siguiente comando:

```
$ check_hss_s6a_certificate ‘‘ruta para guardar el archivo’’ ‘‘nombre del equipo’’
```

En nuestro caso los valores de los parámetros del comando anterior son:

- Ruta del archivo → *"/usr/local/etc/oai/freeDiameter/"*
- Nombre del equipo → *"hss.5GLaboratory"*

Finalmente una vez creados los certificados del HSS, abrimos un terminal y nos situamos en la carpeta `.../openair-cn/SCRIPTS`; primero compilamos y después realizamos su puesta en marcha. Como sigue en los siguientes comandos:

```
$ ./build_hss -c
$ ./run_hss
```

**2. Configuración del MME**

A continuación, una vez en funcionamiento el HSS, seguimos con la modificación del archivo de configuración del MME, el cual se encuentra en la ruta `/usr/local/etc/oai/mme.conf` En la simulación no modificaremos los parámetros de la lista *Globally Unique Mobile Management Entity Identifer* (GUMMEI), ni de la lista TAI, por lo que permanecerán por defecto. Los parámetros que modificaremos de acuerdo con la topología mostrada en la figura 6.4.

- REALM → Nombre de dominio.
- MAXENB → Número máximo de estaciones eNB.
- MAXUE → Número máximo de equipos móviles UE.
- En el apartado S6A
  - S6A CONF → Ruta donde se encuentra la configuración para el interfaz S6A.
  - HSS HOSTNAME → Nombre del equipo HSS.
- En el apartado NETWORK INTERFACES:
  - MME INTERFACE NAME FOR S1 MME → Es la interfaz de conexión con el equipo OAISIM
  - MME IPV4 ADDRESS FOR S1 MME → La dirección IP de la tarjeta con la que nos conectaremos al equipo OAISIM.
  - MME INTERFACE NAME FOR S11 MME → El interfaz para la conexión con el S-GW.
  - MME IPV4 ADDRESS FOR S11 MME → Dirección IP de la tarjeta local para la conexión con el S-GW.
  - MME PORT FOR S11 MME → Puerto para la conexión.

- En el apartado S-GW LIST SELECTION
  - SGW IPV4 ADDRESS FOR S11 → Dirección IP del S-GW.

Los valores que hemos establecido a los parámetros mencionados anteriormente se muestran a continuación:

- REALM = "5GLaboratory";
- MAXENB = "2";
- MAXUE = "16";
- En el apartado S6A:
  - S6A CONF = "/usr/local/etc/oai/freeDiameter/mme_fd.conf";
  - HSS HOSTNAME = "hss";
- En el apartado NETWORK INTERFACES:
  - MME INTERFACE NAME FOR S1 MME = "eth1";
  - MME IPV4 ADDRESS FOR S1 MME = "192.168.31.1/24";
  - MME INTERFACE NAME FOR S11 MME = "lo";
  - MME IPV4 ADDRESS FOR S11 MME = "127.0.11.1/8";
  - MME PORT FOR S11 MME = "2123";
- En el apartado S-GW LIST SELECTION:
  - SGW IPV4 ADDRESS FOR S11 = "127.0.11.2/8";

Otro archivo que modificaremos es el denominado `mme_fd.conf` el cual nos permite modificar los parámetros para la conexión entre el MME y el HSS a través del interfaz S6A, que como sabemos nos proporciona el soporte para la movilidad. En nuestro caso el archivo se encuentra ubicado en la ruta `/usr/local/etc/oai/freeDiameter`, y los parámetros a modificados son los siguientes:

- En el apartado configuración TLS:
  - TLS Cred → Ruta de los archivos para la claves y certificado TLS.
  - TLS CA → Ruta del archivo de la autoridad certificadora.
- En el apartado de punto de conexión:
  - ConnectPeer → Nombre del equipo HSS al que nos conectaremos.
  - ConnectTo → Dirección IP del equipo HSS.
  - Port → Puerto por el que estableceremos la conexión.
  - Realm → Nombre de dominio.

Los valores que hemos establecido en los parámetros anteriores se muestran a continuación:

- En el apartado configuración TLS:

  - TLS Cred = "/usr/local/etc/oai/freeDiameter/mme.cert.pem", "/usr/local/etc/oai/freeDiameter/mme.key.pem";
  - TLS CA = "/usr/local/etc/oai/freeDiameter/mme.cacert.pem";
- En el apartado de punto de conexión:
  - ConnectPeer = "hss.5GLaboratory"
  - ConnectTo = "127.0.0.1"
  - Port = 3868
  - Realm = "5GLaboratory"

La posterior acción a realizar una vez concluida la modificación de los archivos anteriores es crear las claves y certificados de autenticación. Se obtienen del mismo modo que para el HSS, así que, en la terminal nos situaremos en la carpeta `.../openair-cn/SCRIPTS` y ejecutaremos el siguiente comando:

```
$ check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter/ epc.5GLaboratory
```

Finalmente, una vez que tenemos terminada la configuración de los archivos anteriores, procedemos a poner en funcionamiento el MME, primero compilando los archivos y después lanzándolo. Utilizamos los siguientes comandos, y situándonos en la carpeta `.../openair-cn/SCRIPTS`.

```
$ ./build_mme -c
$ ./run_mme
```

### 3. Configuración del S-GW

Por otra parte y para terminar la configuración del equipo que actúa como CN realizaremos la modificación del archivo de configuración para el S-GW, denominado `spgw.conf` y que se encuentra en la ruta `/usr/local/etc/oai`. Al igual que los anteriores, los parámetros que hemos modificado son los siguientes:

- En el apartado NETWORK INTERFACES:
  - SGW INTERFACE NAME FOR S11 → Nombre del interfaz con el que nos conectaremos al MME.
  - SGW IPV4 ADDRESS FOR S11 → Dirección IP del interfaz anterior.
  - SGW INTERFACE NAME FOR S1U S12 S4 UP → Nombre del interfaz por el que recibiremos los datos del usuario, es decir al que nos envían los eNB dichos datos.
  - SGW IPV4 ADDRESS FOR S1U S12 S4 UP → Dirección IP del interfaz anterior.
  - SGW IPV4 PORT FOR S1U S12 S4 UP → Puerto al que recibiremos las conexiones.
- En el apartado IP ADDRESS POOL:
  - IPV4 LIST → Lista de direcciones IP que serán asignadas a los UE.

Los valores de los parámetros son los siguientes:

- En el apartado NETWORK INTERFACES:
  - SGW INTERFACE NAME FOR S11 = “lo”;
  - SGW IPV4 ADDRESS FOR S11 = “127.0.11.2/8”;
  - SGW INTERFACE NAME FOR S1U S12 S4 UP = “eth1”;
  - SGW IPV4 ADDRESS FOR S1U S12 S4 UP = “192.168.1.1/24”;
  - SGW IPV4 PORT FOR S1U S12 S4 UP = “2152”;
- En el apartado IP ADDRESS POOL:
  - IPV4 LIST = “172.16.0.0/12”;

Finalmente, una vez que hemos concluido de modificar los archivos de configuración, debemos de compilar y lanzar el S-GW. Al igual que en los otros casos nos situaremos en la carpeta `.../openair-cn/SCRIPTS`. En este caso no es necesario crear los certificados de autenticación. Los comandos son:

```
$ ./build_spgw -c
$ ./run_spgw
```

### 6.3.4. Configuración de OAISIM

En este apartado procederemos con la configuración de OAISIM, en la máquina virtual OAISIM, tal y como se muestra en la figura 6.4. Además, hemos asumido que se ha realizado la instalación de OAISIM anteriormente.

En primer lugar, lo que debemos hacer es modificar el archivo de configuración, que en nuestro caso se encuentra ubicado en la ruta `.../openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.generic.oaisim.local_mme.conf`. Los parámetros que hemos modificado en este archivo son los siguientes:

- Identificación de celda y del eNB:
  - eNB ID → Identificador del eNB.
  - cell type → Nombre del tipo de celda.
  - eNB name → Nombre del eNB.
- Códigos de área:
  - traking area code → Código de área, debe coincidir con el del MME.
  - mobile country code → Código móvil de país, debe coincidir con el del MME.
  - mobile network code → Código de la red móvil, también debe coincidir con el del MME.
- Parámetros físicos:
  - frame type → Tipo de modulación que utilizaremos.
  - eutra band → Banda usada por el eNB.
  - downlink frequency → Frecuencia para el enlace descendente.

  - uplink frequency offfset → Desplazamiento en frecuencia para el enlace ascendente.
  - tx gain → Ganancia de transmisión en dB???
  - rx gain → Ganancia de recepción en ...
- Parámetros del MME:
  - ipv4 → Dirección IP v4 del MME.
  - preference → Indicamos la preferencia de IP si la versión 4 ó la 6.
- Parámetros de red:
  - ENB INTERFACE NAME FOR S1 MME → Nombre del interfaz de red a través del cual nos conectaremos al MME.
  - ENB IPV4 ADDRESS FOR S1 MME → Dirección IP del interfaz anterior.
  - ENB INTERFACE NAME FOR S1U → Nombre del interfaz de red a través del cual se enviarán los datos del usuario.
  - ENB IPV4 ADDRESS FOR S1U → Dirección IP del interfaz anterior.
  - ENB PORT FOR S1U → Puerto para los datos del usuario.

Los valores que les hemos asignado a cada uno de los parámetros anteriores se muestran a continuación.

- Identificación de celda y del eNB:
  - eNB ID = “0Xe00”;
  - cell type = “CELL MACRO ENB”;
  - eNB name → “eNB 5GLaboratory OAISIM”;
- Códigos de área:
  - traking area code = “1”;
  - mobile country code = “208”;
  - mobile network code = “93”;
- Parámetros físicos:
  - frame type = FDD;
  - eutra band = 7;
  - downlink frequency = 2680000000L;
  - uplink frequency offfset = -120000000;
  - tx gain = “25”;
  - rx gain = “20”;
- Parámetros del MME:
  - ipv4 = “192.168.31.1”;
  - preference = “ipv4”;
- Parámetros de red:

- ENB INTERFACE NAME FOR S1 MME = “eth1”;
- ENB IPV4 ADDRESS FOR S1 MME = “192.168.1.2/24”;
- ENB INTERFACE NAME FOR S1U = “eth1”;
- ENB IPV4 ADDRESS FOR S1U = “192.168.1.2/24”;
- ENB PORT FOR S1U = “2152”;

Seguidamente, una vez concluida la configuración, nos situaremos en la carpeta `.../openairinterface5g/cmake-targets/`. Compilaremos con el siguiente comando, y los parámetros se detallan a continuación.

```
$ ./build_oai -c --oaisim --UE -x
```

El significado de los parámetros añadidos es:

- -c → Borra los archivos temporales para realizar de nuevo la compilación.
- –oaisim → Crea el simulador de OAISIM.
- –UE → Crea las partes específicas del UE.
- -x → Agrega la opción de osciloscopio por software.

Cuando hemos dado por terminada la compilación, lanzaremos la aplicación. Hemos escogido el modelo que recrea un eNB y un UE virtualizados utilizando la interfaz S1. Para ello nos situaremos en la carpeta `.../openairinterface5g/cmake-targets/tools` y escribimos el siguiente comando:

```
$ sudo -E ./run_enb_ue_virt_s1
```

## 6.4. OAI EPC + OAI eNB (USRP B210) + UE

En esta parte del capítulo, realizaremos la implementación del primer escenario real. En éste utilizaremos el EPC de OAI, el eNB de OAI combinado con un USRP B210, y dos UE reales. En la figura 6.5 obtenemos un plano visual del escenario montado en el laboratorio.

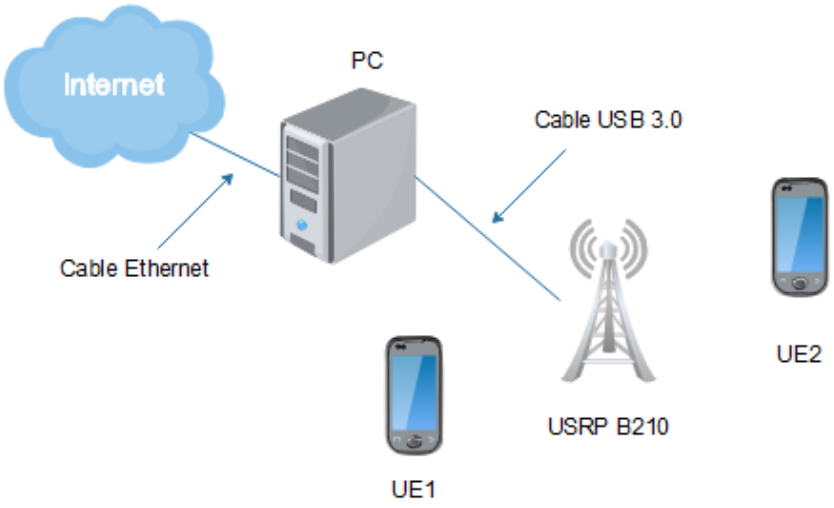



Figura 6.5: Escenario real con OAI EPC + OAI eNB (USRP B210) + UE.

### 6.4.1. Preparando los equipos

En primer lugar, y siguiendo los pasos que hicimos durante el apartado de simulación, prepararemos los equipos reales antes de descargar e instalar los directorios de OAI correspondientes. Los pasos que debemos realizar y que se explicaron anteriormente, son los que se muestran a continuación:

- Instalación del sistema operativo Ubuntu 14.04 LTS (64 bits).
- Instalación de kernel v.4.7 de baja latencia o superior.
- Desactivación de los estados C para la gestión de energía.
- Desactivación del escalado en frecuencia.

Una vez que hemos realizado todas las recomendaciones de la lista anterior, continuamos con la configuración de conectividad entre el PC en el que instalaremos OAI-CN y OAI eNB, tal y como muestra la figura 6.6.

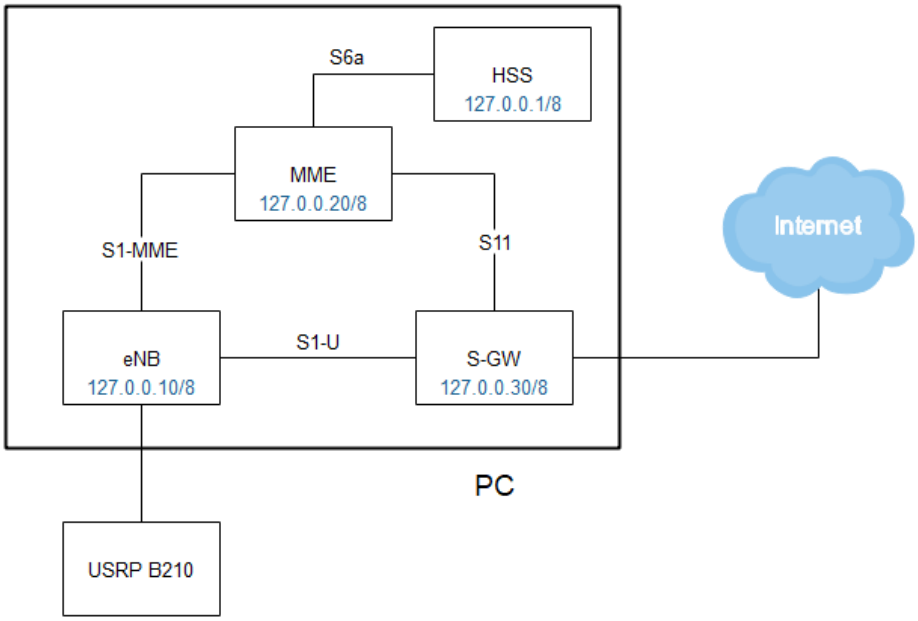



Figura 6.6: Estructura lógica del escenario real.

### 6.4.2. Configurando el OAI *Core Network* (CN)

En este apartado configuraremos todas las entidades que componen el EPC, como ya vimos anteriormente en el apartado de *Simulación*.

En primer lugar comenzaremos con la introducción de datos en la base de datos que utilizará el HSS para la autenticación de los UE. Dichos parámetros se recogen en las tablas 6.4, 6.5 y 6.6.

| **USIM** | 11 |
|---|---|
| **IMSI** | 208920100001100 |
| **ICCID** | 8988211000000142102 |
| **ACC** | 0001 |
| **PIN1** | 1008 |
| **PUK1** | 41798190 |
| **PIN2** | 5915 |
| **PUK2** | 53546826 |
| **Ki** | 8baf473f2f8fd09487cccbd7097c6862 |
| **OP** | 1006020f0a478bf6b699f15c062e42b3 |
| **OPC** | - |
| **ADM1** | 85601364 |
| **KIC1** | C12232C5E86752D0597196D27F640636 |
| **KID1** | 82D43091335F66F90AC2B68B2353CD0C |
| **KIK1** | B6C20DFF86230E0DAE7AB6E78AD948AF |
| **MSISDN** | 34612345611 |

Tabla 6.4: Parámetros de las tarjetas USIM 11.

| **USIM** | 12 |
|---|---|
| **IMSI** | 208920100001101 |
| **ICCID** | 8988211000000142110 |
| **ACC** | 0002 |
| **PIN1** | 5028 |
| **PUK1** | 11316366 |
| **PIN2** | 7434 |
| **PUK2** | 97477116 |
| **Ki** | 8baf473f2f8fd09487cccbd7097c6862 |
| **OP** | 1006020f0a478bf6b699f15c062e42b3 |
| **OPC** | - |
| **ADM1** | 10022674 |
| **KIC1** | 55E61E4A504269064C2ADA87B9B4D99D |
| **KID1** | 4B1868D945A52C4C15BA7E65DD3D6026 |
| **KIK1** | B873B267A9EDD2B8AC52781B6C6968DC |
| **MSISDN** | 34612345612 |

Tabla 6.5: Parámetros de las tarjetas USIM 12.

| **USIM** | 13 |
|---|---|
| **IMSI** | 208920100001102 |
| **ICCID** | 8988211000000142128 |
| **ACC** | 0004 |
| **PIN1** | 5055 |
| **PUK1** | 54612055 |
| **PIN2** | 0820 |
| **PUK2** | 64274346 |
| **Ki** | 8baf473f2f8fd09487cccbd7097c6862 |
| **OP** | 1006020f0a478bf6b699f15c062e42b3 |
| **OPC** | - |
| **ADM1** | 00415661 |
| **KIC1** | 104F29DB9FFBF0490B40BA6E85D8447B |
| **KID1** | C3F474929296F43BC09A247990B79763 |
| **KIK1** | 2DAC10833325E11F9BB075A154EAD650 |
| **MSISDN** | 34612345613 |

Tabla 6.6: Parámetros de la tarjeta USIM, 13.

Utilizaremos tres tarjetas USIM programables, a las cuales les introduciremos los datos de las tablas anteriores (más adelante explicaremos el proceso). Por otro lado, para incluir los datos de las tarjetas a utilizar en la base de datos, primero activaremos el servicio *mysql* en el equipo en cuestión (visto en el apartado de *simulación*). Una vez activo el servicio, accederemos a él mediante el siguiente comando, y además debemos de importar la base de datos `oai_db.sql` si comprobamos que *mysql* no la detecta.

```
$ mysql -u root -p
```

Una vez dentro, le indicaremos que nos muestre todas las bases de datos para asegurarnos de que nuestra base en cuestión es detectada, tal y como se muestra en la figura 6.7.

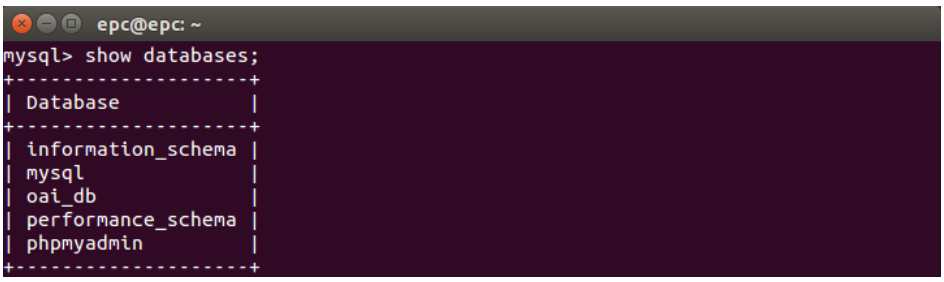



Figura 6.7: Bases de datos en el EPC.

El siguiente paso, es seleccionar la base de datos con el comando ‘‘`use oai_db;`’’, y pedirle que nos muestre las tablas que contiene, con ‘‘`show tables;`’’. En la figura 6.8 observamos las tablas que alberga.


Figura 6.8: Tablas en la base *oai_db*.

El propósito de cada una de las tablas de la figura anterior es el siguiente:

- Tabla *apn* : guarda información relacionada con los APN.
- Tabla *mmeidentity* : contiene la información correspondiente al MME.
- Tabla *pdn* : almacena principalmente los parámetros de la asociación entre un subscriptor (IMSI) y un APN, además parámetros de la QoS.
- Tabla *pgw* : guarda información relacionada con el P-GW.
- Tabla *terminal-info* : tiene información adicional relacionada con el IMEI.
- Tabla *users* : en esta tabla se almacena la información del subscriptor como es el IMSI, IMEI, $K_i$, etc.

Las tablas que en nuestro caso nos interesan son: *mmeidentity, pdn, pgw, users.*

**Tabla *mmeidentity*.** En esta tabla ingresaremos los datos mostrados en la tabla 6.7 ,de la misma forma que lo vimos en el apartado de simulación.

| idmmeidentity | mmehost | mmerealm | UE-Reachability |
|---|---|---|---|
| 1 | mme.OpenAir5G.Alliance | OpenAir5G.Alliance | 0 |

Tabla 6.7: Valores para la tabla *mmeidentity*.

**Tabla *pdn*.** En dicha tabla ingresaremos los datos mostrados en la tabla 6.8; y el comando para ingresar los valores se muestra justo después de ella. De entre los parámetros interesantes que podemos destacar son:

- *apn:* nombre del pundo de acceso.
- *pdn_type:* tipo de PDN, que actualmente sólo está disponible en IPv4.
- *aggregate_ambr_ul:* umbral para la máxima velocidad permitida en el enlace ascendente, en Mbps.
- *aggregate_ambr_dl:* umbral para la máxima velocidad permitida en el enlace descendente, en Mbps.
- *pgw_id:* identificador del P-GW en la tabla *pgw*.
- *users_imsi:* IMSI asociado a estos parámetros.

Las entradas que deberán aparecer en la tabla deberán tener los siguientes parámetros:

| Parámetros | USIM 11 | USIM 12 | USIM 13 |
|---|---|---|---|
| **id** | 22 | 23 | 24 |
| **apn** | ltebox | ltebox | ltebox |
| **pdn_type** | IPv4 | IPv4 | IPv4 |
| **pdn_ipv4** | 0.0.0.0 | 0.0.0.0 | 0.0.0.0 |
| **pdn_ipv6** | 0:0:0:0:0:0:0:0 | 0:0:0:0:0:0:0:0 | 0:0:0:0:0:0:0:0 |
| **aggregate_ambr_ul** | 50000000 | 50000000 | 50000000 |
| **aggregate_ambr_dl** | 100000000 | 100000000 | 100000000 |
| **pgw_id** | 3 | 3 | 3 |
| **users_imsi** | 208920100001100 | 208920100001101 | 208920100001102 |
| **qci** | 9 | 9 | 9 |
| **priority_level** | 15 | 15 | 15 |
| **pre_emp_cap** | DISABLED | DISABLED | DISABLED |
| **pre_emp_vul** | ENABLED | ENABLED | ENABLED |
| **LIPA_Permissions** | LIPA-only | LIPA-only | LIPA-only |

Tabla 6.8: Valores para la tabla *pdn*.

El siguiente comando servirá de ejemplo para introducir los valores de los parámetros mostrados en la tabla *pdn*.

```
$ mysql > INSERT INTO pdn (`id`, `apn`, `pdn_type`, `pdn_ipv4`, `pdn_ipv6`, `
    aggregate_ambr_ul`, `aggregate_ambr_dl`, `pgw_id`, `users_imsi`, `qci`, `
    priority_level`,`pre_emp_cap`,`pre_emp_vul`, `LIPA-Permissions`) VALUES ('111',  '
    oai.ipv4','IPV4', '0.0.0.0', '0:0:0:0:0:0:0:0', '50000000', '100000000', '3',
    '208920100001100', '9', '15', 'DISABLED', 'ENABLED', 'LIPA-ONLY');
```

**Tabla *pgw*.** En esta tabla ingresaremos los valores mostrados en la tabla 6.9.

| id | ipv4 | ipv6 |
|---|---|---|
| 1 | 127.0.0.1 | 0:0:0:0:0:0:0:1 |

Tabla 6.9: Valores de la tabla *pgw*.

El comando para ingresar los valores es:

```
$ mysql > INSERT INTO pgw (‘id‘, ‘ipv4‘, ‘ipv6‘) VALUES (’1’, ’127.0.0.1’,
    ’0:0:0:0:0:0:0:1’);
```

**Tabla *users*.** Esta tabla contiene la principal información asociada al cliente. La información que ingresaremos en los parámetros será la existente en las tablas 6.4 y 6.6.

Uno de los valores que no introduciremos nosotros manualmente es el del OPC, ya que lo calcula automáticamente mediante la fórmula:

$$OPC = AES_{128}(k_i, OP) \oplus OP$$

El comando para introducir los valores es el siguiente. Aquí se muestra el equivalente para introducir los datos de la primera tarjeta USIM.

```
$ mysql > INSERT INTO users (‘imsi‘, ‘msisdn‘, ‘imei‘, ‘imei_sv‘, ‘ms_ps_status‘, ‘
    rau_tau_timer‘, ‘ue_ambr_ul‘, ‘ue_ambr_dl‘, ‘access_restriction‘, ‘mme_cap‘, ‘
    mmeidentity_idmmeidentity‘, ‘key‘, ‘RFSP-Index‘, ‘urrp_mme‘, ‘sqn‘, ‘rand‘, ‘OPc‘)
     VALUES (’208920100001100’, ’34612345611’, NULL, NULL, ’PURGED’, ’120’,
    ’50000000’, ’100000000’, ’47’, ’0000000000’, ’7’, 0x8baf473f 2
    f8fd09487cccbd7097c6862, ’1’, ’0’, ’’, 0x00000000000000000000000000000000, ’’);
```

### Configuración del HSS

Una vez que hemos finalizado la introducción de información en la base de datos, en primer lugar realizaremos la modificación del archivo de configuración para poder lanzar el HSS. Como vimos en el apartado 6.3, nos situaremos en el directorio que contiene el archivo `hss.conf`.

En este archivo los parámetros que modificaremos junto con sus valores se muestran en la siguiente lista:

- *MYSQL_server* : “127.0.0.1”;
- *MYSQL_user* : “root”;
- *MYSQL_pass* : “linux”;
- *MYSQL_db* : “oai_db”;
- *FD_conf* : “/usr/local/etc/oai/freeDiameter/hss_db.conf”;

Debemos tener en cuenta que en esta versión de OAI descargada, en el archivo anterior tiene que estar comentado el parámetro *"OPERATOR_key"*, ya que lo comprueba en la base de datos.

El siguiente archivo a modificar es `hss_db.conf` en el cual estableceremos el dominio, el nombre del equipo, las rutas para los certificados de autenticación, y los puertos de escucha entre otros parámetros. Los parámetros junto con los valores son:

- *Identity* = "hss.OpenAir5G.Alliance";
- *Realm* = "OpenAir5G.Alliance";
- *TLS_Cred* = /usr/local/etc/oai/freeDiameter/hss.cert.pem", /usr/local/etc/oai /freeDiameter/hss.key.pem";
- *TLS_CA* = /usr/local/etc/oai/freeDiameter/hss.cacert.pem";
- *Port* = 3868;
- *SecPort* = 5868;
- *LoadExtension* = "acl_wl.fdx" : "usr/local/etc/oai/freeDiameter/acl.conf"

En este archivo al igual que ocurría con el anterior debe estar comentado el parámetro *"ConnectPeer"*, ya que en esta versión de repositorio se ha solucionado la localización por nombres.

Por último, en el archivo `acl.conf` modificaremos el parámetro: *ALLOW_OLD_TLS* ∗.OpenAir5G.Alliance , para permitir la conexión con equipos del mismo dominio; también se indica que se acepta el intercambio CER/CEA no protegido con Inband-Security-Id = TLS.

Una ver terminado la modificación de los archivos, necesitamos crear los certificados para el HSS, por lo que usaremos el mismo comando que en el apartado 6.3. Nos situamos en la carpeta `.../openair-cn/SCRIPTS` y en el terminal ejecutaremos el comando:

```
$ check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.OpenAir5G.Alliance;
```

Para lanzar el HSS, escribimos los comandos:

```
$ ./build_hss -c
$ ./run_hss
```

### Configurando el MME

En este subapartado, realizaremos la configuración de los archivos del MME para poder lanzarlo y que funcione correctamente. Primero, modificaremos el archivo `mme.conf` que se encuentra en el mismo directorio que el archivo `hss.conf`. Los parámetros que hemos modificado son los siguientes:

- *REALM* = "OpenAir5G.Alliance";
- *S6A_CONF* = "/usr/local/etc/oai/freeDiameter/mme_fd.conf";

- *HSS_HOSTNAME* = "hss";
- *GUMMEI_LIST* = ({MCC="208"; MNC="92"; MME_GID="4"; MME_CODE="1"; });
- *TAI_LIST* = ({MCC="208"; MNC="92"; TAC = "1";});
- *MME_INTERFACE_NAME_FOR_S1_MME* = "lo";
- *MME_IPV4_ADDRESS_FOR_S1_MME* = "127.0.0.20/8";
- *MME_INTERFACE_NAME_FOR_S11_MME* = "lo";
- *MME_IPV4_ADDRESS_FOR_S11_MME* = "127.0.0.20/8";
- *MME_PORT_FOR_S11_MME* = 2123;
- *SGW_IPV4_ADDRESS_FOR_S11* = "127.0.0.30/8";

A continuación, modificaremos el archivo `mme_fd.conf`, en los que deberemos revisar los valores que tienen el nombre del equipo, el dominio, la dirección de las claves para la autenticación, y el puerto entre otros. Nuestros valores en los parámetros son:

- *Identity* = "epc.OpenAir5G.Alliance";
- *Realm* = "OpenAir5G.Alliance";
- *TLS_Cred* = "/usr/local/etc/oai/freeDiameter/mme.cert.pem", "/usr/local/etc/oai/freeDiameter/mme.key.pem";
- *TLS_CA* = "/usr/local/etc/oai/freeDiameter/mme.cacert.pem";
- *Port* = 3870;
- *SecPort* = 5870;
- *ConnectPeer* = "hss.OpenAir5G.Alliance" { ConnectTo = "127.0.0.1"; No_SCTP ; No_IPv6; Prefer_TCP; No_TLS; port = 3868; realm = "OpenAir5G.Alliance";};

En los parámetros anteriores observamos que el puerto por el que se conectará el MME con el HSS coincide en los archivos de configuración. Además debemos de asegurarnos que los dominios están escritos correctamente, al igual que las direcciones IP.

Seguidamente, el próximo paso es crear las para la autenticación y autorización para la conexión entre ambos equipos (MME y HSS). De la misma manera que para el HSS, pero utilizando el *script* correspondiente. El comando a escribir en la terminal es:

```
$ check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter/ mme.OpenAir5G.Alliance;
```

Para lanzar el MME, escribimos en la terminal los comandos:

```
$ ./build_mme -c
$ ./run_mme
```

### Configurando el S-GW

Para finalizar la puesta en marcha de todos los equipos que componen EPC, vamos a realizar la modificación del archivo de configuración del S-GW el cual es `spgw.conf`. Los parámetros que modificaremos son:

- *SGW_INTERFACE_NAME_FOR_S11* = "lo";
- *SGW_IPV4_ADDRESS_FOR_S11*= "127.0.0.30/8";
- *SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP* = "lo";
- *SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP* = "127.0.0.30/8";
- *SGW_IPV4_PORT_FOR_S1U_S12_S4_UP* = 2152;
- *PGW_INTERFACE_NAME_FOR_SGI* = "enp3s0";
- *PGW_MASQUERADE_SGI* = "yes";
- *IPV4_LIST* = ("172.16.0.0/12");

Una vez finalizada la modificación de dicho archivo, realizaremos la compilación y el lanzamiento del S-GW con los siguientes comandos, dando por finalizada la configuración del OAI CN.

```
$ ./build_spgw -c
$ ./run_spgw
```

## 6.4.3. Configurando el OAI eNB

En esta parte realizaremos la configuración de la entidad que actúa como eNB. El *hardware* que utilizaremos como estación base es el USRP B210 el cual irá conectado mediante USB 3.0 a dicho equipo.

En primer lugar, descargaremos el repositorio correspondiente para el eNB de OAI. Debemos instalar la herramienta *git* si no la tenemos. Para descargar los archivos correspondientes, abrimos un terminal y nos situamos en la carpeta donde queremos descargarlo y ejecutamos el siguiente comando para clonar dicho repositorio:

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

En segundo lugar, y una vez finalizada la descarga de los archivos nos situaremos en la carpeta `openairinterface5g`. Ahora, realizaremos por primera y única vez la instalación de unos paquetes que son necesarios para que OAI eNB funcione correctamente, únicamente debemos ejecutar el siguiente comando:

```
$ ./cmake_targets/build_oai -I
```

En tercer lugar, ya terminada la instalación del *software* adicional necesario para el eNB, realizaremos la modificación del archivo de configuración que es utilizado para el USRP B210. Éste se encuentra en la ruta `.../openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf`, y los parámetros que modificaremos, de acuerdo con nuestro mapa lógico de red, son:

- Códigos numéricos:
  - tracking area code → Código de área de seguimiento.
  - mobile country code → Código de identificación de país.
  - mobile network code → Código de identificación del operador.
- Parámetros físicos:
  - frame type → Tipo de trama a utilizar (FDD ó TDD)
  - eutra band → Banda de frecuencias seleccionada.
  - downlink frequency → Frecuencia portadora para el enlace descendente.
  - uplink frequency offset → Desviación de la frecuencia en el enlace ascendente respecto a la frecuencia del enlace descendente.
  - N RB DL → Radio bloques en el enlace de bajada.
  - tx gain → Ganancia de transmisión en dBm.
  - rx gain → Ganancia de recepción en dBm.
- Parámetros MME:
  - ipv4 → Dirección IP del equipo que ejecuta el MME.
  - preference → Preferencia por direcciones IP v4 ó v6.
- Apartado interfaces de red:
  - ENB INTERFACE NAME FOR S1 MME → Nombre del interfaz por el que nos conectaremos al MME.
  - ENB IPV4 ADDRESS FOR S1 MME → Dirección IP del interfaz por el que nos conectaremos al MME.
  - ENB INTERFACE NAME FOR S1U → Nombre del interfaz por el que se transmitirán los datos del UE.
  - ENB IPV4 ADDRESS FOR S1U → Dirección IP del interfaz por el que se transmitirán los datos del UE.
  - ENB PORT FOR S1U → Puerto para el interfaz S1U.

Los valores que hemos establecido en los parámetros mencionados anteriormente son:

- Códigos numéricos:
  - tracking area code = “1”;
  - mobile country code = “208”;
  - mobile network code = “92”;
- Parámetros físicos:

- frame type = “FDD”;
- eutra band = “7”;
- downlink frequency = 2685000000L;
- uplink frequency offset = -120000000;
- N RB DL = 25;
- tx gain = 90;
- rx gain = 120;

- Parámetros MME:
  - ipv4 = “127.0.0.20”;
  - preference = “ipv4”;
- Apartado interfaces de red:
  - ENB INTERFACE NAME FOR S1 MME = lo
  - ENB IPV4 ADDRESS FOR S1 MME = “127.0.0.10/24”;
  - ENB INTERFACE NAME FOR S1U = lo
  - ENB IPV4 ADDRESS FOR S1U = “127.0.0.10/24”;
  - ENB PORT FOR S1U = “2152”;

Terminada la configuración de los parámetros del USPR B210, realizaremos la compilación de los archivos con el siguiente comando:

```
$ ./cmake_targets/build_oai -c -w USRP --eNB
```

El significado de los parámetros adicionales es el siguiente:

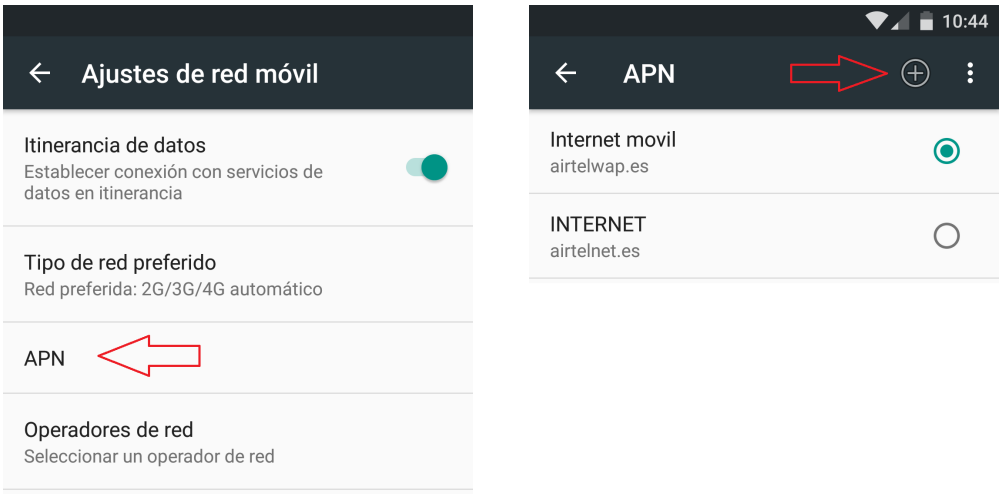
- -c → Elimina todos los archivos de compilaciones anteriores.
- -w → Indica el soporte *hardware* que vamos a utilizar.
- –eNB → Crea el *softmodem* de LTE.

Finalmente, cuando se haya terminado la compilación, lanzaremos la aplicación. Para ello nos situaremos en la carpeta `openairinterface5g`, y en el ejecutable debemos indicar el archivo de configuración deseado. En nuestro caso hemos escrito el siguiente comando:

```
$ ./cmake_targets/lte_build_oai/build/lte-softmodem -d -O ./targets/PROJECTS/GENERIC-
    LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf
```

### 6.4.4. Configurando el UE

Por otra parte, los dispositivos que también hemos configurado para realizar el análisis del escenario real, son los dispositivos móviles utilizados. La configuración que debemos realizar es crear un *Access Point Name* (APN); para ello seleccionaremos en el menú principal *Ajustes/Redes móviles/APN*. Podemos ver los pasos para crear un APN en la figura 6.9



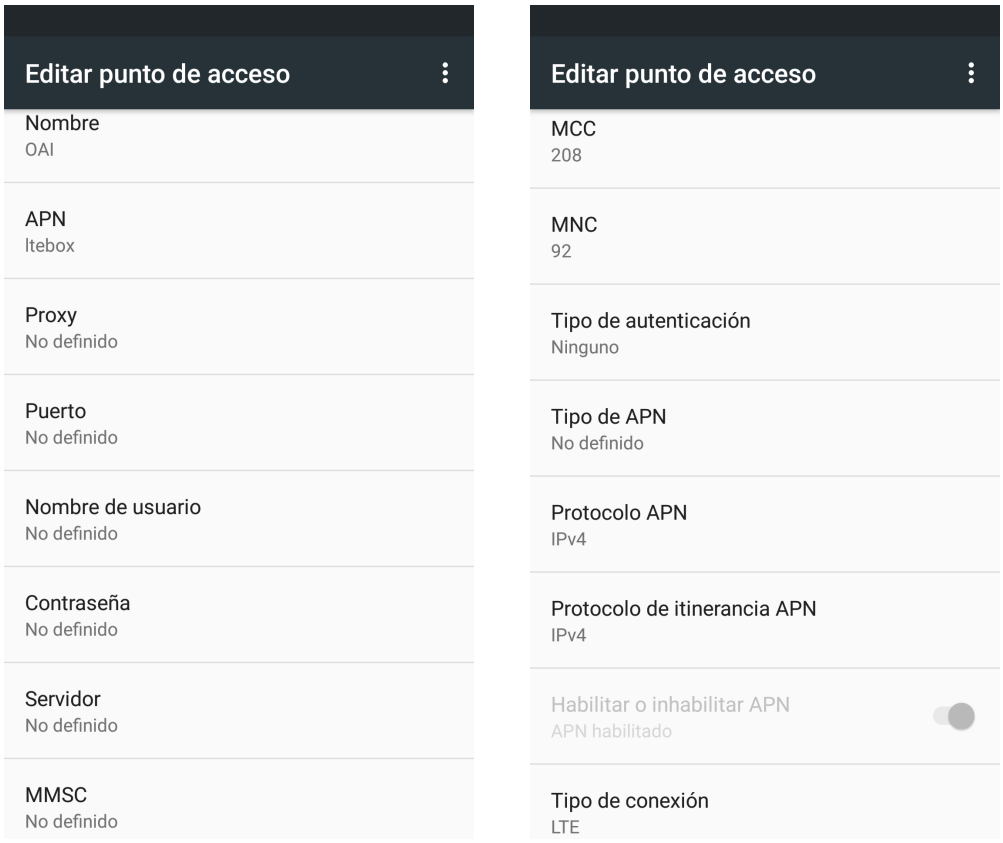

(a) Selección APN (b) Crear APN

Figura 6.9: Pasos para crear un nuevo APN.

Los parámetros que configuraremos para dicho APN, con sus respectivos valores son los siguientes:

- Nombre : OAI
- APN : ltebox
- MCC : 208
- MNC : 92
- Tipo de autenticación : Ninguno
- Protocolo APN : IPv4
- Protocolo de itinerancia : IPv4
- Tipo de conexión : LTE

La configuración establecida en el dispositivo se puede apreciar en la figura 6.10, una vez hecho ésto la guardamos. Finalmente para comprobar que se ha realizado correctamente la configuración, el nuevo APN aparecerá en la lista de la figura 6.9 b).




(a) Parámetros APN (b) Cont. Parámetros APN

Figura 6.10: Configurando el APN.

### 6.4.5. Programación de las tarjetas USIM

En este apartado se detalla la programación de las tarjetas USIM de SYSMOCOM, vistas en el capítulo 3, y concretamente en el apartado 3.3.1.

Los datos que introduciremos para realizar la programación de las tarjetas USIM se encuentran en las tablas 6.4 y 6.6.

El programa con el que se realiza la configuración de estas tarjetas está desarrollado en Python, y además requiere que el SO del equipo sea Linux. El archivo de programación lo podemos descargar directamente desde la página http://git.osmocom.org/pysim/.

Antes de realizar la programación de las tarjetas, debemos de tener a mano el valor de los parámetros mostrados en la tabla 6.10, con sus respectivos valores. Los datos aquí mostrados son los correspondientes para la programación de la tarjeta USIM 11.

| Descripción | Parámetro | Valor |
|---|---|---|
| Número PC/SC del acceso a USIM | -p | 1 |
| Tipo de tarjeta USIM | -t | sysmoUSIM-SJS1 |
| PIN de administrador | -a | 85601364 |
| Mobile Country Code | -x | 208 |
| Mobile Network Code | -y | 92 |
| IMSI | -i | 208920100001100 |
| ICCID | -s | 8988211000000142102 |
| OPC | –op | 1006020f0a478bf6b699f15c062e42b3 |
| $K_i$ | -k | 8baf473f2f8fd09487cccbd7097c6862 |

Tabla 6.10: Valores para la programación de la tarjeta USIM 11.

Un ejemplo de cómo debemos introducir estos parámetros en la consola de comandos de Linux sería el siguiente, que corresponde con la programación de la tarjeta anteriormente mencionada (USIM 11). Antes de realizar la programación, debemos tener la tarjeta USIM insertada en nuestro Lector de tarjetas y éste conectado a nuestro PC mediante el cable USB. Nos situaremos en el directorio donde se encuentra el programa previamente descargado, `pySim-prog.py`; y escribimos la siguiente orden:

```
$ ./pySim-prog.py -p 1 -t sysmoUSIM-SJS1 -a 85601364 -x 208 -y 92 -i 208920100001100
-s 8988211000000142102 --op=1006020f0a478bf6b699f15c062e42b3
-k 8baf473f2f8fd09487cccbd7097c6862
```

Si se ha realizado correctamente la programación de nuestra tarjeta nos saldrá un mensaje en la consola como éste:

```
Generated card parameters :
> Name    : Magic
> SMSP    : e1ffffffffffffffffffffffff0581005155f5ffffffffffff000000
> ICCID   : 8988211000000142102
> MCC/MNC : 208/92
> IMSI    : 208920100001100
> Ki      : 8baf473f2f8fd09487cccbd7097c6862
> OPC     : e734f8734007d6c5ce7a0508809e7e9c
> ACC     : None

Programming ...
Done !
```

# Capítulo 7

# Análisis

## 7.1. Introducción

En este capítulo, en primer lugar realizaremos un análisis de los mensajes de señalización que se llevan a cabo en el escenario de simulación, visto detalladamente en el capítulo 6. Posteriormente, realizaremos un análisis del escenario real que se ha creado a partir de un PC de sobremesa, un USRP B210, y dos teléfonos móviles realizando el rol de UEs. El montaje de éste último escenario también se ha detallado en el capítulo 6. Se realizará un análisis de la capa física, de los mensajes intercambiados en la señalización de la red LTE, y finalmente unos test de velocidad.

## 7.2. EPC + OAISIM

En la figura 7.1 se ha obtenido el flujo de mensajes intercambiados entre las máquinas virtuales OAISIM y EPC. Se han capturado con Wireshark y los hemos filtrado con el protocolo S1-AP para quedarnos exclusivamente con los mensajes de señalización.

### 7.2.1. Señalización

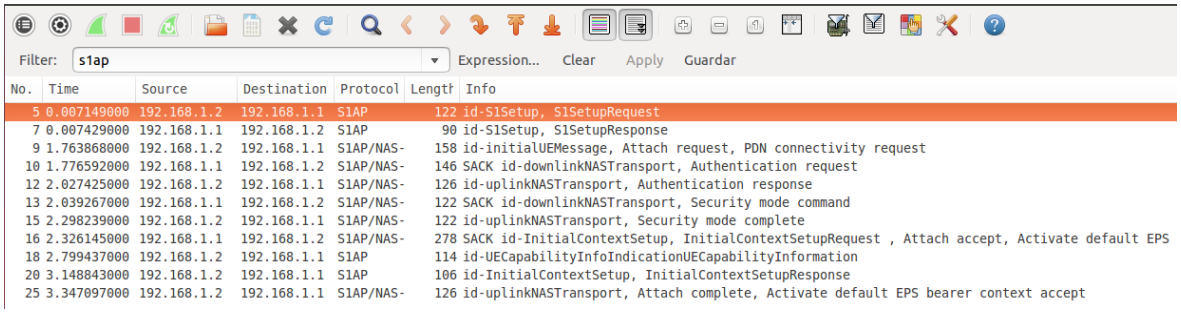



Figura 7.1: Mensajes de señalización entre OAISIM y OAI EPC.

A continuación, explicaremos la función de cada uno de los mensajes destacando alguna información que podemos encontrar dentro de ellos, como puede ser la dirección IP del MME, los identificadores de las sesiones para los UE, o la información intercambiada para la autenticación. Cabe destacar que en el apartado 7.3 se llevará a cabo una mayor explicación en el contenido de estos mensajes, así como la extensión de éstos.

### 7.2.2. Conexión OAISIM eNB - MME

La primera secuencia de mensajes que nos encontramos son los mostrados en la figura 7.2, que se envían desde el eNB que inicia la conexión hacia el MME, y viceversa para establecer la conexión del eNB a la red LTE.

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 5 | 0.007149000 | 192.168.1.2 | 192.168.1.1 | S1AP | 122 | id-S1Setup, S1SetupRequest |
| 7 | 0.007429000 | 192.168.1.1 | 192.168.1.2 | S1AP | 90 | id-S1Setup, S1SetupResponse |

Figura 7.2: Mensajes para la conexión del eNB a la red.

En el primer mensaje de la estación base, el *S1 Setup Request*, se envía diversa información desde la identidad global del eNB como es el MCC y el MNC, hasta el propio nombre de la estación base.

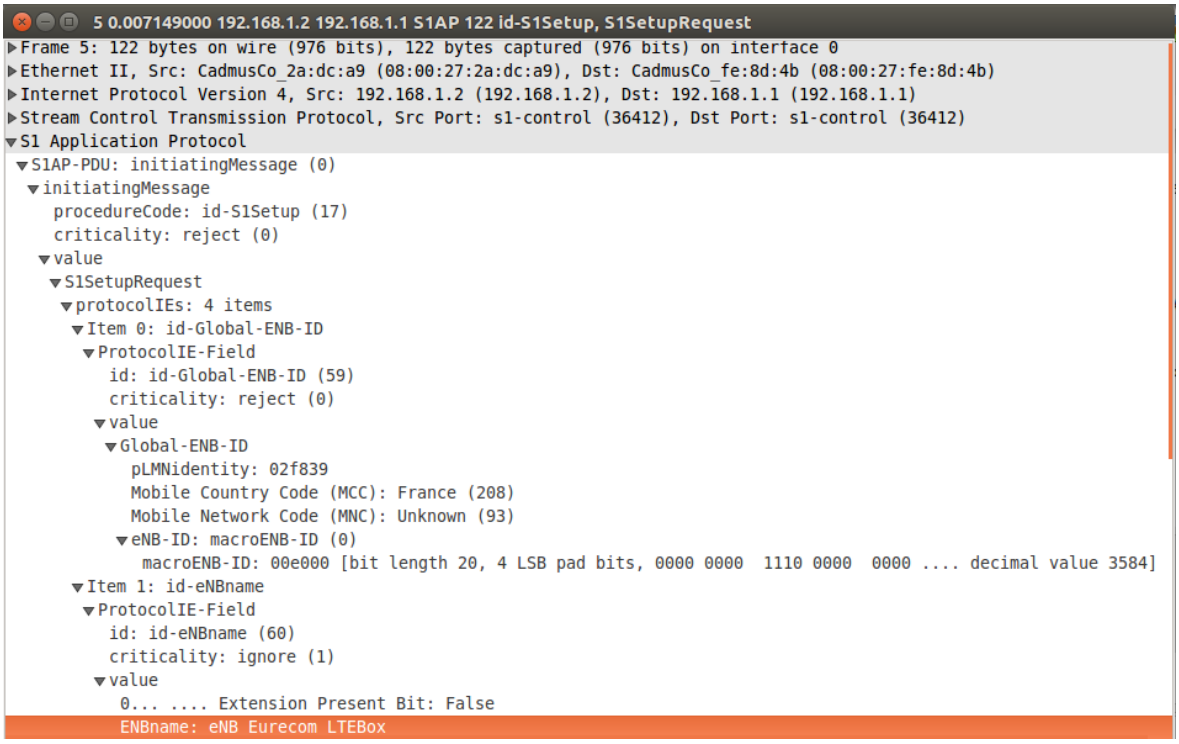



Figura 7.3: Parte del mensaje S1 Setup Request.

### 7.2.3. Conexión OAISIM UE - MME

El siguiente mensaje es el *Initial UE Message*, que es enviado al MME y una vez que se ha recibido, el MME le asigna un identificador para esta sesión con el UE, y tenerlo así localizado. Se puede ver parte de la información de dicho mensaje en la figura 7.4. En él se incluyen datos como el identificador de conexión entre el eNB y el UE, el IMSI del usuario en concreto que quiere conectarse a la red, los protocolos soportados para el cifrado y la autenticación, parámetros que identifican a la red como el MCC, el MNC, etcétera.


Figura 7.4: Parte del mensaje Initial UE Message.

### 7.2.4. Autenticación y Seguridad

Una vez enviado el mensaje *Initial UE Message*, el MME inicia el procedimiento de autenticación y el modo de seguridad. Estos mensajes se muestran en la figura 7.5. En primer lugar, se envía el mensaje *Authentication Request* hacia el eNB, y éste lo reenviará hacia el UE.

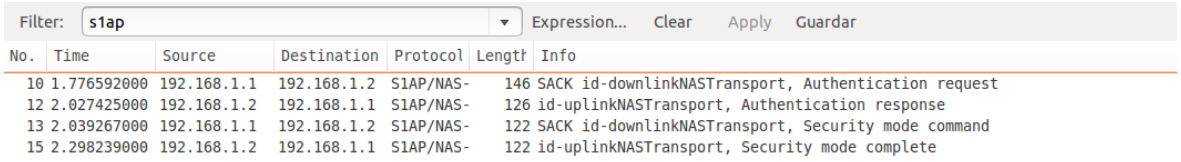



Figura 7.5: Intercambio de mensajes para la autenticación y la seguridad.

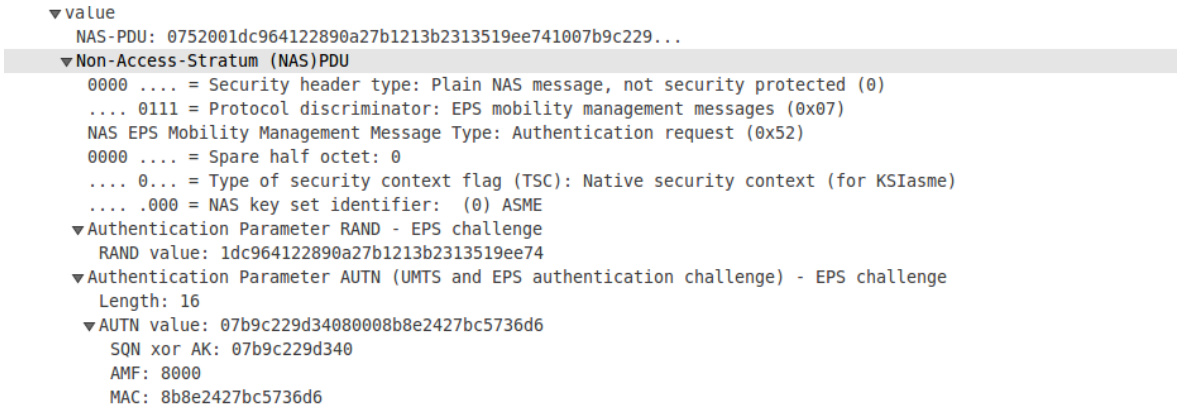



Figura 7.6: Valores de los parámetros RAND y AUTN.

En el mensaje *Authentication Request* se envía al UE el desafío con el parámetro RAND y el parámetro AUTN, para iniciar el procedimiento de autenticación. Una vez

recibido este mensaje el UE enviará la respuesta del desafío con el parámetro RES en el mensaje *Authentication Response*. Una vez finalizada la parte de autenticación, se envía el mensaje *Security Mode Command* por parte del MME hacia el UE, y el mensaje *Security Mode Complete*, que se utilizan para ordenar al UE la activación de la seguridad AS. La seguridad AS engloba la integridad de la señalización, el cifrado de ésta. Por otra parte, también engloba la integración y el cifrado de los datos del plano de usuario.

### 7.2.5. Contexto inicial

Una vez terminados los procedimientos de autenticación y seguridad, el MME envía el mensaje *Initial Context Setup* hacia el UE. En éste mensaje se incluye información referente con las especificaciones para la conexión del UE en la red. Contiene la dirección IP del S-GW (192.168.1.1), la dirección IP asignada al UE (en nuestro caso es 172.16.0.2/12), las direcciones IP de los servidores DNS, el valor de la calidad de servicio que recibirá el usuario, el GUTI, y parámetros para la velocidad de tráfico en los enlaces ascendente y descendente, etcétera.

```
16 2.326145000 192.168.1.1 192.168.1.2 S1AP/NAS-EPS 278 SACK id-InitialContextSetup, InitialContextSetupRequest ,
    transportLayerAddress(IPv4): 192.168.1.1 (192.168.1.1)
  gTP-TEID: 00000001
  nAS-PDU: 276008369401074202490620 02f839000100385201c10109...
▼ Non-Access-Stratum (NAS)PDU
    0010 .... = Security header type: Integrity protected and ciphered (2)
    .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
    Message authentication code: 0x60083694
    Sequence number: 1
    0000 .... = Security header type: Plain NAS message, not security protected (0)
    .... 0111 = Protocol discriminator: EPS mobility management messages (0x07)
    NAS EPS Mobility Management Message Type: Attach accept (0x42)
    0000 .... = Spare half octet: 0
    .... 0... = Spare bit(s): 0x00
    .... .010 = Attach result: Combined EPS/IMSI attach (2)
  ▶ GPRS Timer - T3412 value
  ▶ Tracking area identity list - TAI list
  ▼ ESM message container
      Length: 56
    ▼ ESM message container contents: 5201c1010909036f6169046970763405 01ac1000025e04fe...
        0101 .... = EPS bearer identity: EPS bearer identity value 5 (5)
        .... 0010 = Protocol discriminator: EPS session management messages (0x02)
        Procedure transaction identity: 1
        NAS EPS session management messages: Activate default EPS bearer context request (0xc1)
      ▶ EPS quality of service
      ▶ Access Point Name
      ▼ PDN address
          Length: 5
          0000 0... = Spare bit(s): 0x00
          PDN type: IPv4 (1)
          PDN IPv4: 172.16.0.2 (172.16.0.2)
```

Figura 7.7: Fragmento del mensaje Initial Context Setup.

A dicho mensaje se responderá con el mensaje *Initial Context Setup Response*, indicando la dirección IP del eNB en el que se encuentra el UE.

## 7.3. OAI EPC + OAI eNB + UE

En esta sección, vamos a realizar un análisis del escenario real mostrado en la figura 7.8. En el cual, como podemos apreciar, tenemos instalado y funcionando el OAI EPC, OAI eNB y dos teléfonos móviles que tendrán el papel de UE.

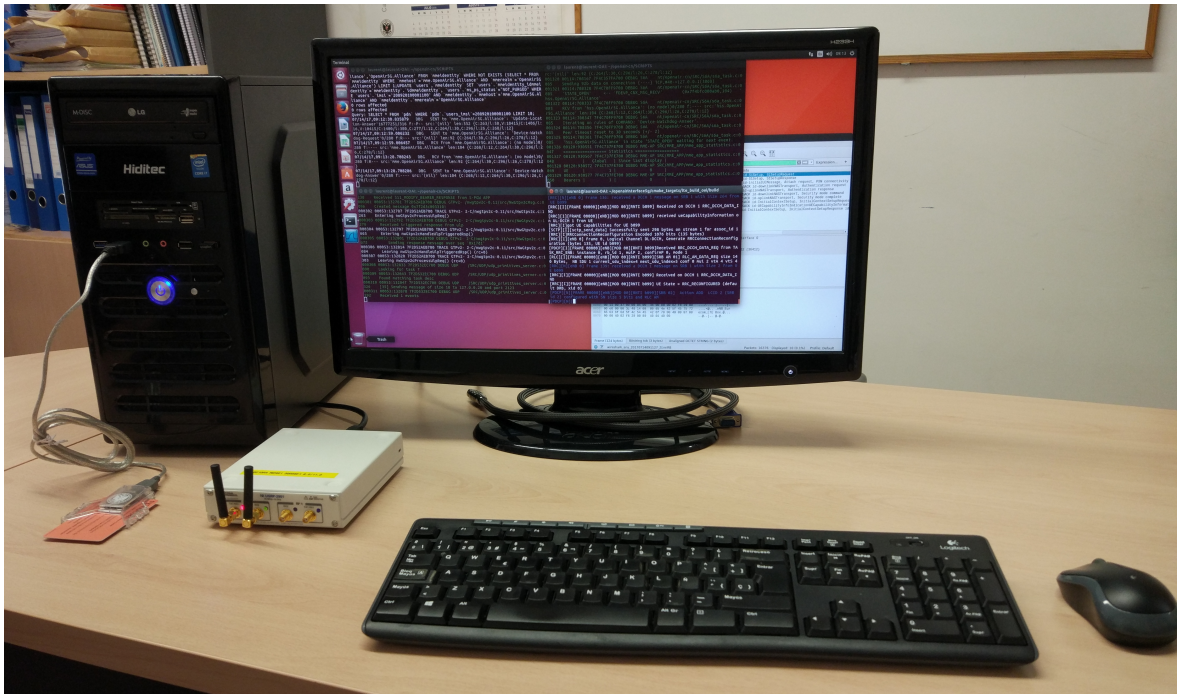



Figura 7.8: Aspecto del escenario real.

A continuación, se detalla un breve resumen con las características físicas de los distintos escenarios evaluados.

- Parte radio.
  - Frecuencia central:
    - UL: 2,565 GHz
    - DL: 2,685 GHz
  - Nº de RB: 25, 50, 100
  - Ancho de banda:
    - UL: 5 MHz, 10 MHz, 20 MHz
    - DL: 5 MHz, 10 MHz, 20 MHz

### 7.3.1. Parte radio

En este apartado evaluaremos la capa física, examinando el ancho de banda, ganancias, slot temporales, la *Power Spectral Density* (PSD) y el espectrograma, tanto en el enlace descendente como en el ascendente. Para ello hemos llevado a cabo varias veces la misma prueba, en la que mediante la aplicación *"SpeedAnalytics"* se ha realizado un test de velocidad utilizando los teléfonos móviles disponibles para el proyecto.

El análisis de la parte radio se ha realizado mediante el analizador de espectros *AGILENT N9010A*, y con scripts del *toolbox* LTE de MATLAB.

### Analizador de espectros

Mientras se realizaban los test de velocidad, con el analizador de espectros conseguimos capturar el ancho de banda que se utilizaba para el enlace ascendente, y así poder realizar varias medidas. En la figura 7.9 observamos la frecuencia central del enlace ascendente (2.564 GHz), verificando así el funcionamiento de los parámetros establecidos por nosotros previamente en el OAI eNB.

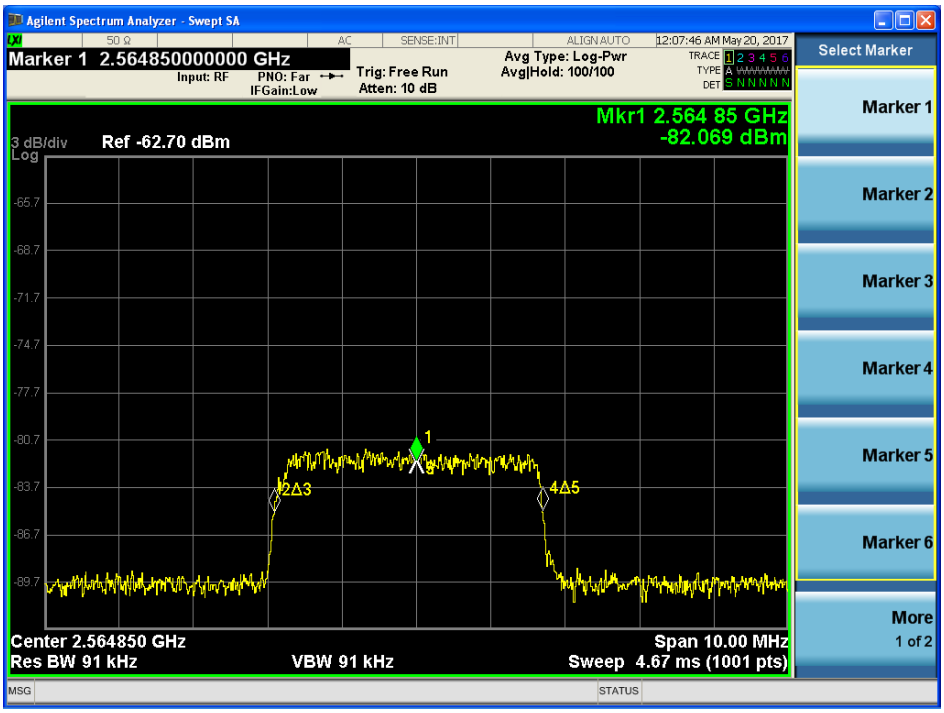



Figura 7.9: Frecuencia central del enlace ascendente.

En la figura 7.10, hemos posicionado el segundo marcador para medir el límite inferior del ancho de banda. Se ha colocado el cursor a -3.298 dbm respecto del pico más alto (ver figura 7.9), y tenemos en la mitad inferior un ancho de 1.91 MHz.

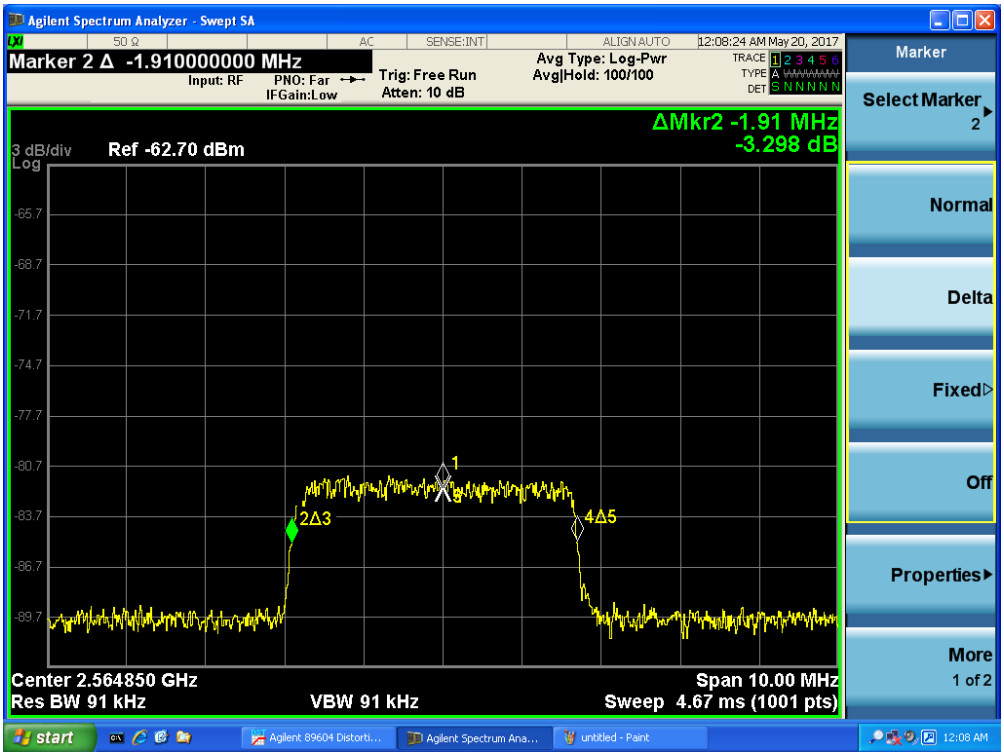



Figura 7.10: Límite inferior del ancho de banda en UL.

En la figura 7.11, hemos procedido de la misma manera posicionando un cursor en la caída de 3.201 dbm, para medir la mitad superior del ancho de banda utilizado. En este caso, tenemos 1.70 MHz.

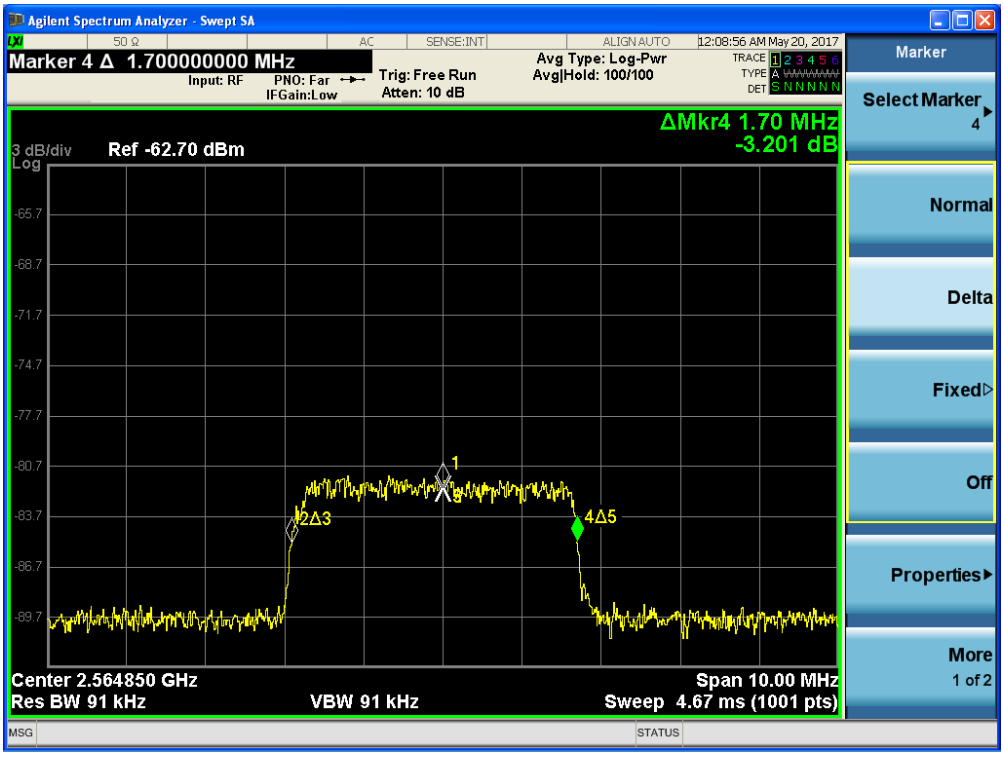



Figura 7.11: Límite superior del ancho de banda en UL.

Si sumamos el tamaño de las dos mitades, vistas en las figuras 7.10 y 7.11, hacen un total de 3.61 MHz. Este resultado depende del dispositivo móvil. Una respuesta a este resultado, es que de acuerdo a las pruebas que estamos realizando con los dispositivos móviles, éstos detecten que no es necesario utilizar todo el ancho de banda para transmitir en el enlace ascendente. Esto hace que no utilice todas las portadoras en dicho canal.

A continuación, se muestran los resultados obtenidos habiéndose realizado la misma prueba pero en este caso nos centraremos en el enlace descendente. En la figura 7.12, observamos el ancho de banda utilizado para el enlace descendente y un marcador en el límite inferior, teniendo como frecuencia central del enlace descendente 2,685 GHz. Teniendo en cuenta los mismos parámetros que para el enlace ascendente, se ha marcando la frecuencia 2.68274 GHz como límite inferior.

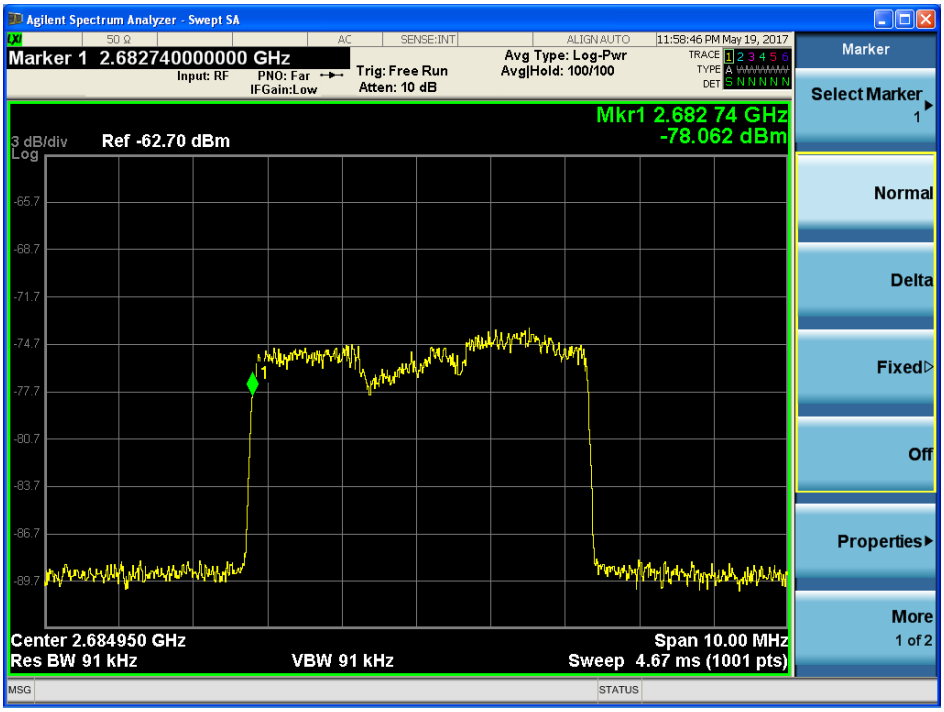



Figura 7.12: Límite inferior del ancho de banda en DL.

En la figura 7.13, de nuevo se ha marcado el límite superior obteniendo la frecuencia de 2.68726 GHz.

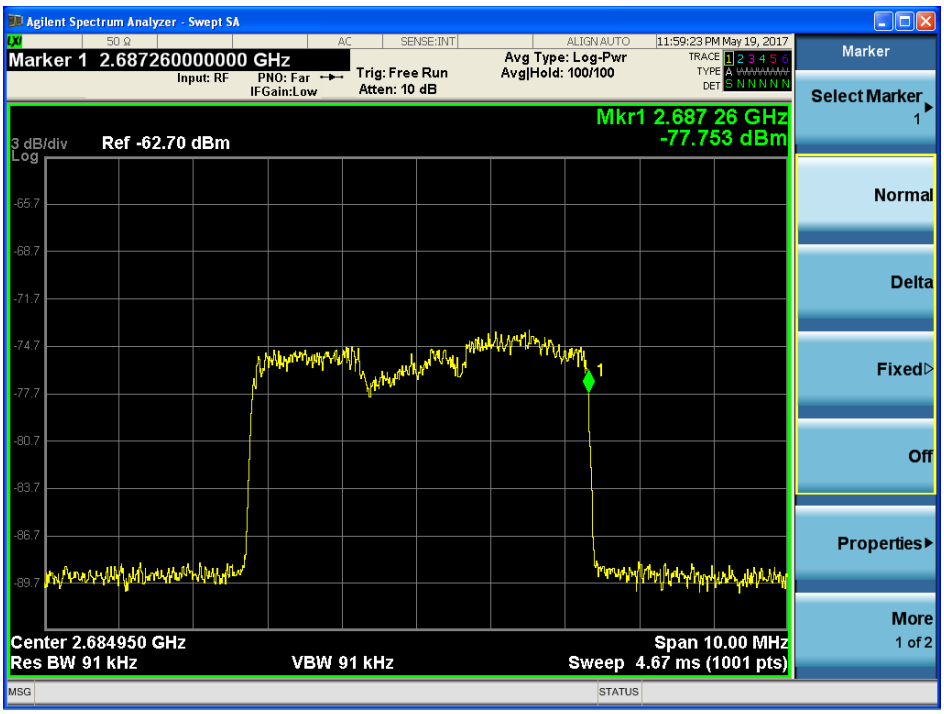



Figura 7.13: Límite superior del ancho de banda en DL.

Una vez obtenidos ambos límites, podemos comprobar que OAI eNB para el enlace descendente utiliza 4.52 MHz. Para DL sí cumple el estándar establecido del 3GPP para el ancho de banda ocupada para la transmisión, teniendo en cuenta que debe tener 5 MHz de anchura nominal. Si realizamos los cálculos para comprobar que el ancho de banda es correcto, tenemos: $25RB * 180kHz = 4,5MHz$, si le añadimos los 250 kHz de banda de guarda tanto en el límite inferior como en el superior, tenemos los 5 MHz de ancho de banda nominal, tal y como se especifica en el estándar. Por lo que el resultado obtenido es correcto.

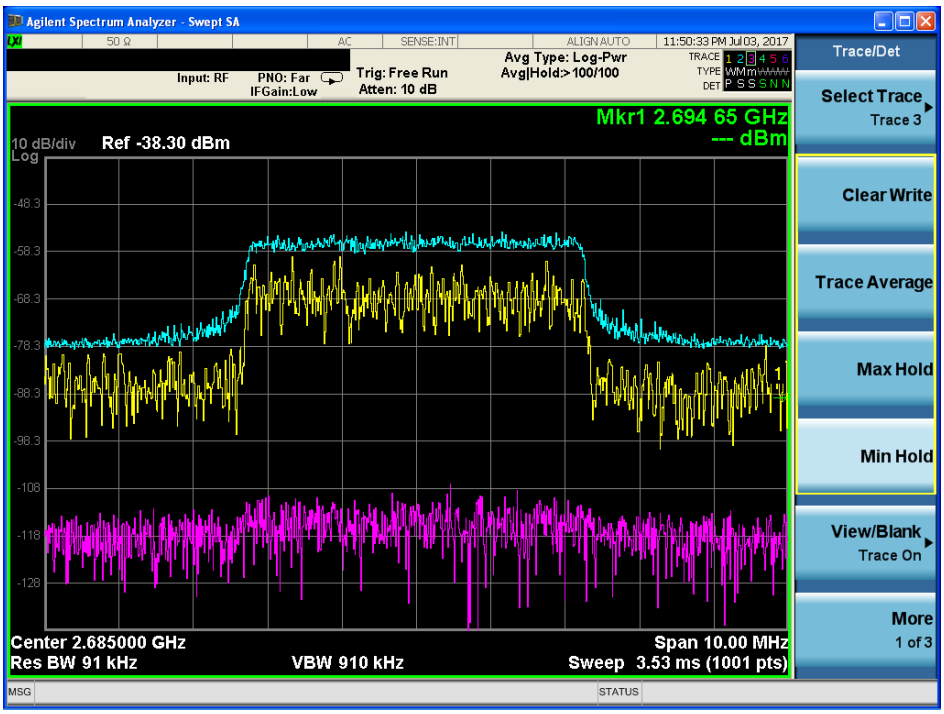



Figura 7.14: Máximos y mínimos de la señal el frecuencia.

Estableciendo un ancho de banda de 5 MHz para el canal descendente, en la figura 7.14 observamos los máximos (color azul) y mínimos (color rosa) de las frecuencias para cada una de las portadoras que componen el canal.

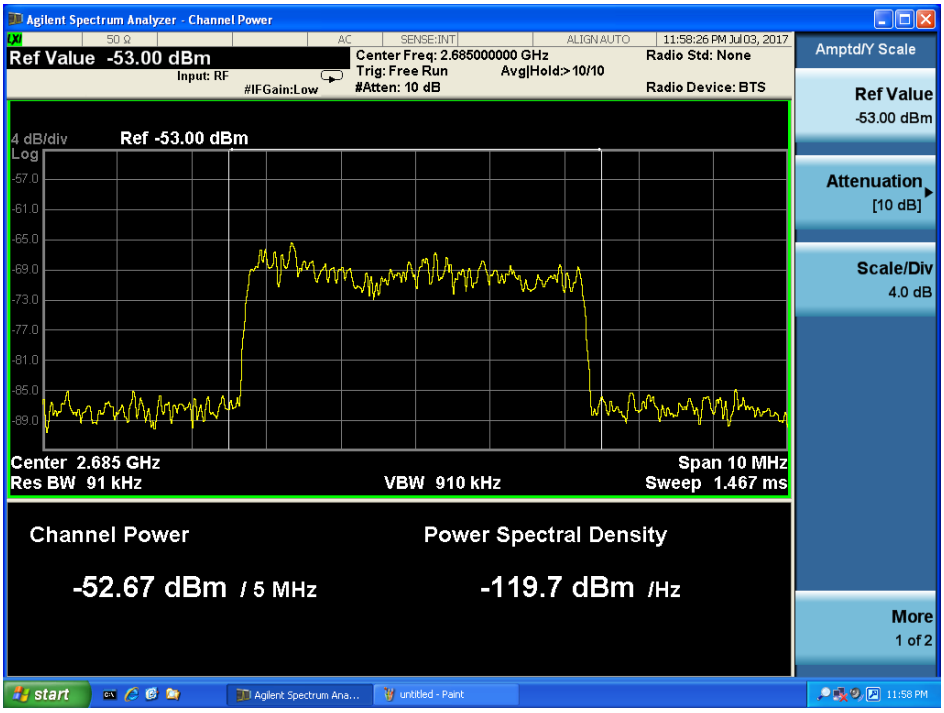



Figura 7.15: PSD en DL y potencia por canal.

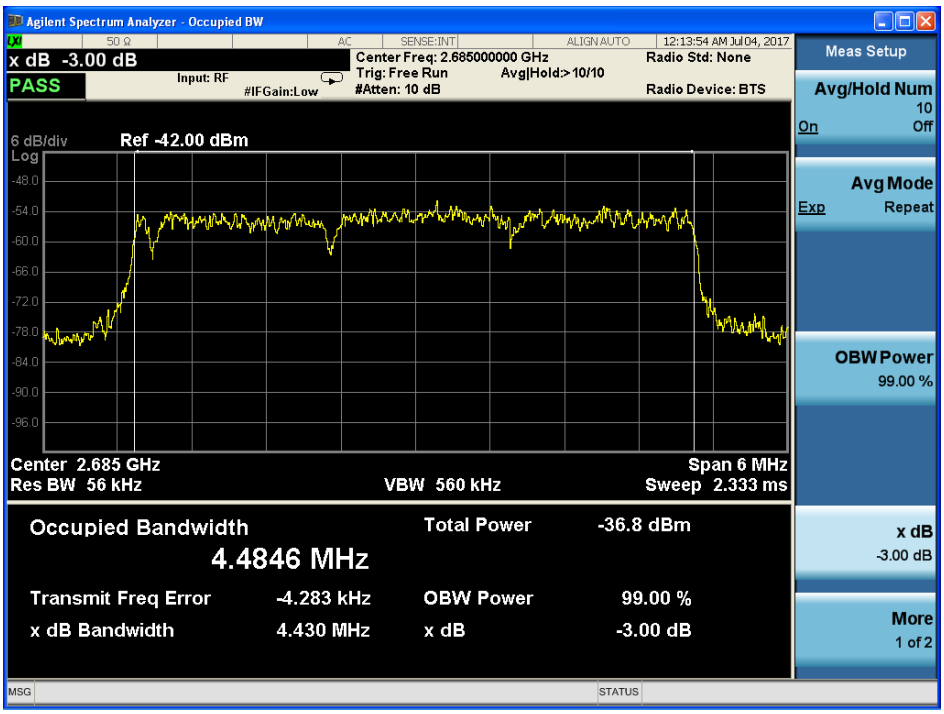



Figura 7.16: BW ocupado al 99 % y en -3 dB.

A continuación, en la figura 7.15 se ha obtenido la densidad de potencia espectral para un ancho de banda de 5MHz, cuyo resultado ha sido de 119,7 dBm/Hz y con una potencia por canal de -52,67 dBm. Este valor tan bajo se debe a que la potencia de transmisión de nuestro eNB es pequeña ya que no deseamos provocar molestias a usuarios en esta misma frecuencia o en frecuencias vecinas (como operadores móviles) con interferencias a la hora de realizar las pruebas; además el analizador de espectros se encuentra muy próximo a la estación base, para obtener mejores resultados.

En la figura 7.16 se ha obtenido el ancho de banda ocupado para dos criterios distintos. El primero de ellos es el BW ocupado al 99 %, obteniendo como resultado 4,4846 MHz; el segundo criterio es en la caída de 3 dB respecto al máximo, cuyo resultado es de 4,43 MHz. Como cabe esperar con el segundo criterio se ha obtenido un valor menor. Además, nos muestra el error de la frecuencia de transmisión ($\pm 4,283 kHz$), el cual se encuentra en torno a 1 ppm. Según la especificación [1], el máximo rango permitido sería para una *Home BS* con un valor de $\pm 0,25 ppm$. Esto es debido a las características propias del USRP empleado como estación base, que no nos permite un error menor. Para disminuir esta deriva, se podría utilizar una señal de referencia de reloj externa que se puede introducir al USRP, o bien utilizar una sincronización basada en un receptor GPS. En cualquier caso, es un resultado muy similar a los 4,5 MHz esperados por la especificación (5 MHz con las bandas de guarda).

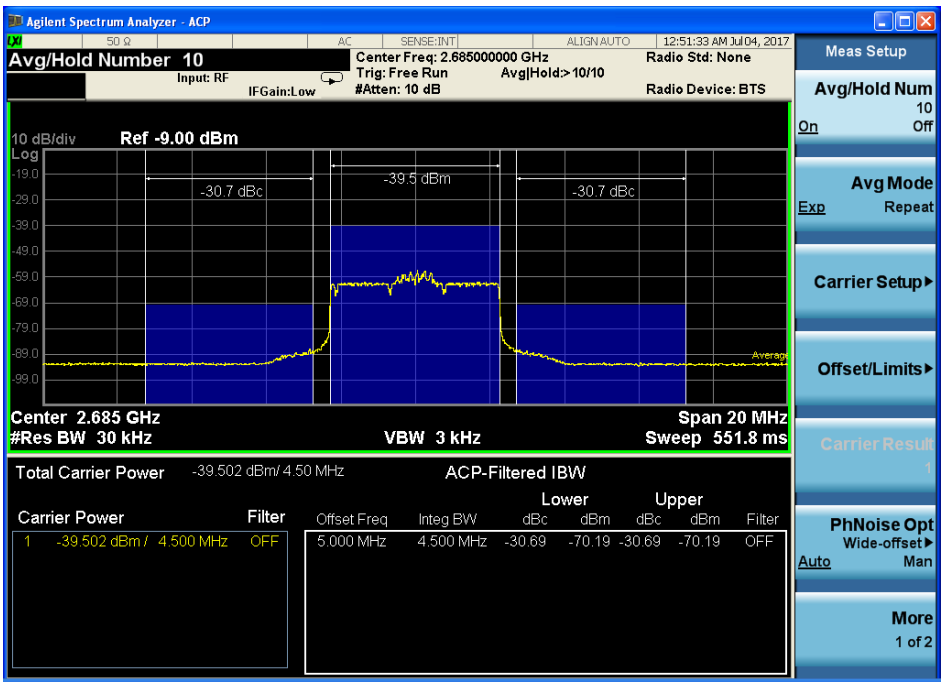



Figura 7.17: Medición del Adjacent Channel Power (ACP).

En la figura 7.17 se ha conseguido medir la potencia de los canales portadores adyacentes a nuestro canal de transmisión, denominados comúnmente canales de desplazamiento. En la especificación [1] del 3GPP se define el *Adjacnt Channel Leakage Ratio* (ACLR) para los distintos anchos de banda especificados para LTE. Se requiere un mínimo de 45dB, mientras que en nuestro caso tenemos una diferencia de 30dB, no cumpliendo en este caso la especificación.

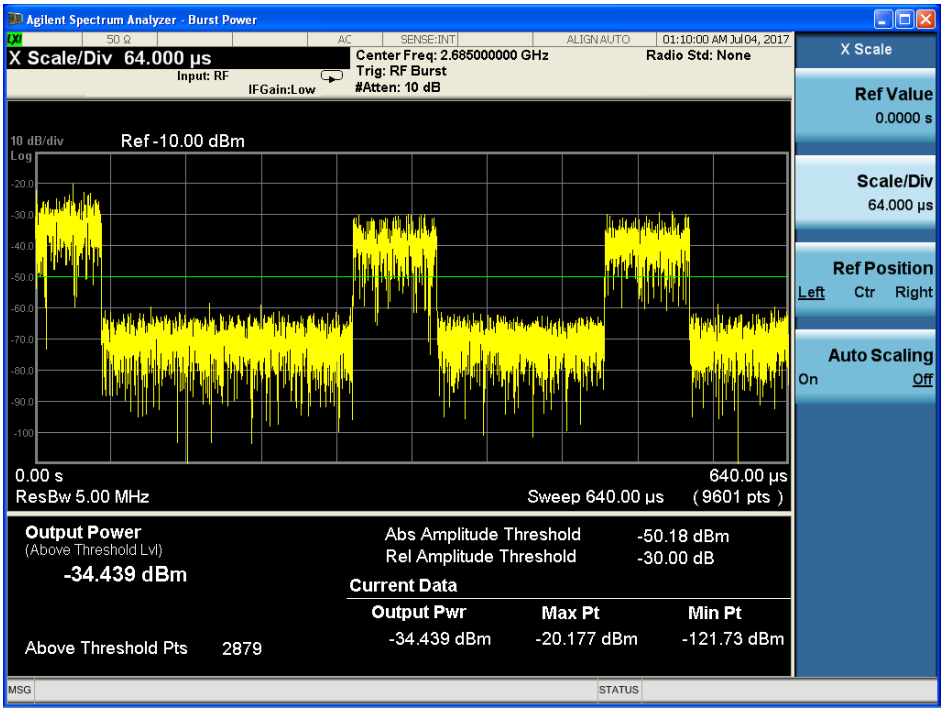



Figura 7.18: Poténcia / tiempo de ráfagas individuales.

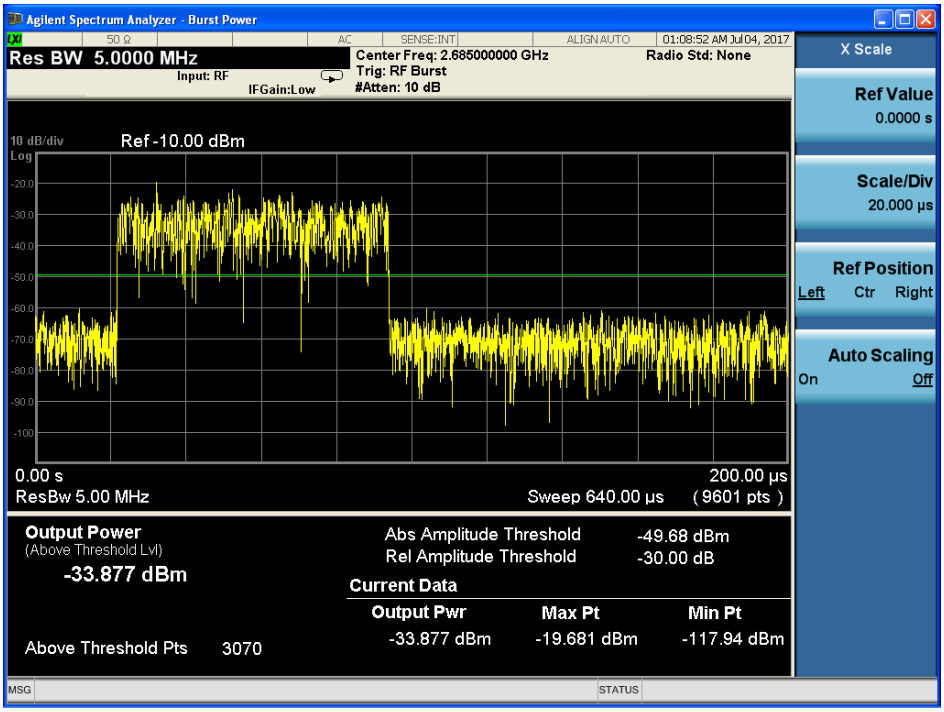
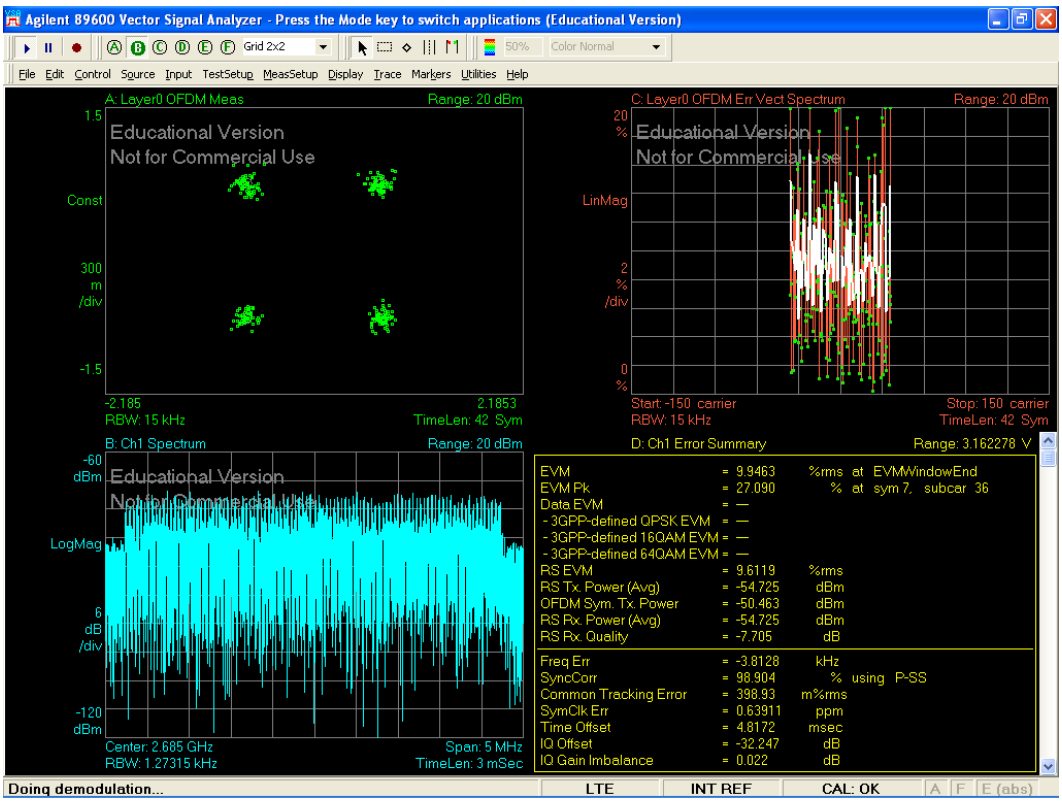
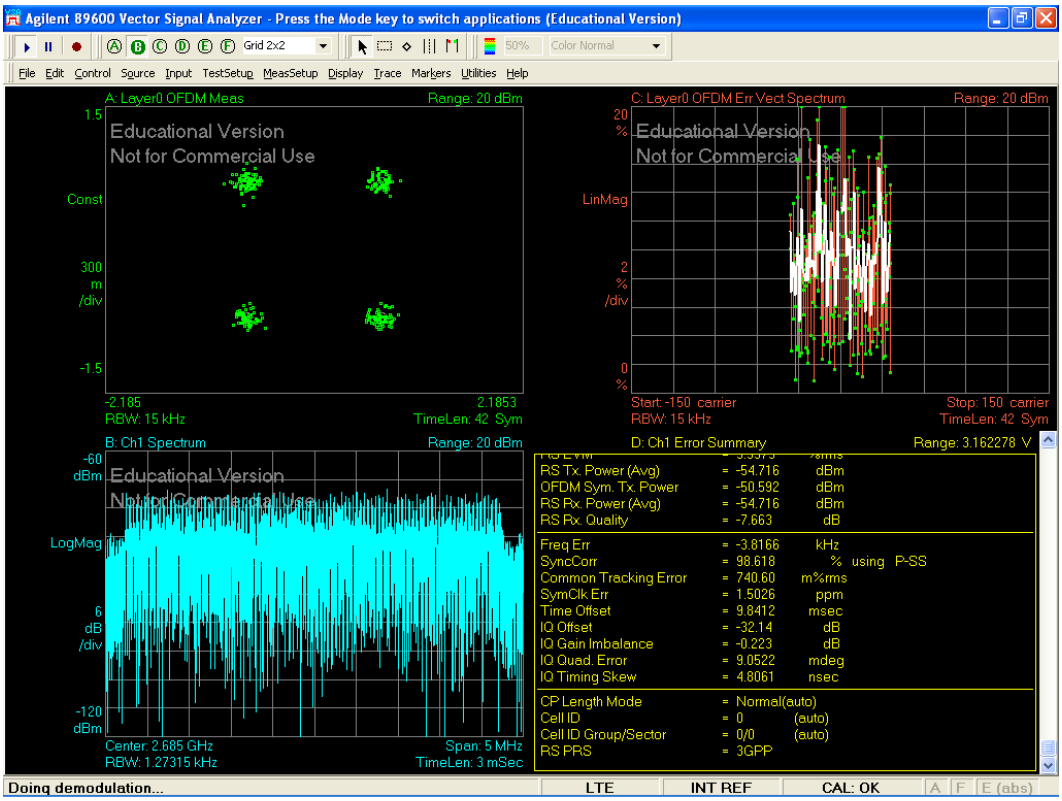



Figura 7.19: Potencia / tiempo de una ráfaga individual.

En las figuras 7.18 y 7.19 podemos observar ráfagas de transmisión por parte del eNB. Según la especificación [1] el periodo de transitorio del transmisor debe ser menor a 17 $\mu$s por lo que el sistema cumple perfectamente la especificación.




(a) Visión 1.




(b) Visión 2.

Figura 7.20: Análisis del canal PBCH.

El analizador de espectros AGILENT N9010A tiene instalado un paquete básico (versión educación) para la demodulación de señales LTE. Debemos indicarle la fre-

cuencia en la que está transmitiendo el eNB, y el canal que nos interesa analizar. En la figura 7.20 se puede apreciar el análisis que realiza para el canal PBCH. En la esquina superior izquierda se encuentra la constelación QPSK que utiliza dicho canal; en el lado opuesto se encuentra el EVM para los símbolos OFDM transmitidos; en la parte inferior izquierda nos encontramos con el espectro de nuestro canal de transmisión (en este caso estamos transmitiendo con 25 RB, que hacen un total de ancho de banda de 5 MHz); y finalmente en la esquina inferior derecha nos muestra un resumen con las características del canal seleccionado, en el que podemos ver entre otras cosas el EVM, la potencia de transmisión para el símbolo OFDM, o el error de frecuencia.

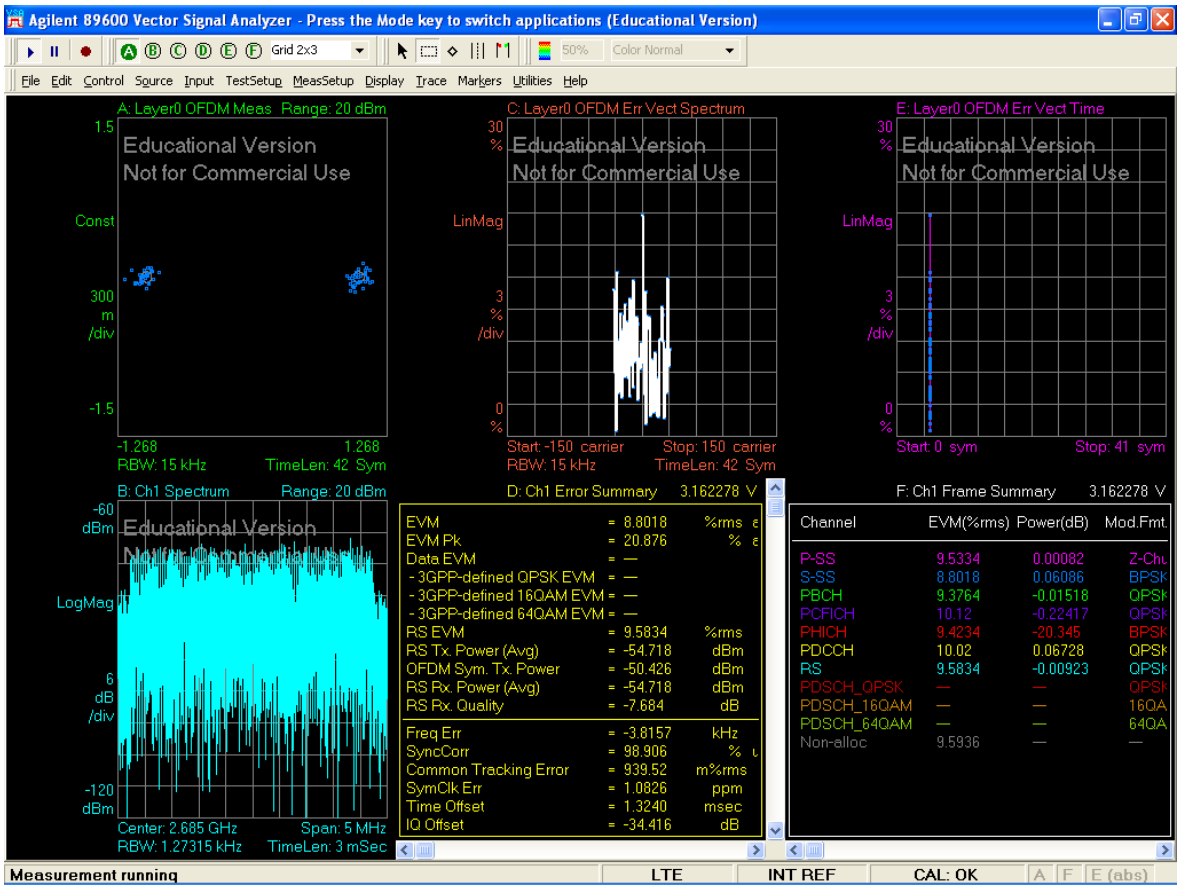



Figura 7.21: Análisis de la señal P-SS.

Con la misma organización de ventanas que la figura anterior, en la figura 7.21 hemos obtenido la demodulación para la señal de sincronismo P-SS. Esta señal utiliza la modulación QPSK, utilizando las secuencias de *Zadoff-Chu*, como veremos más adelante.

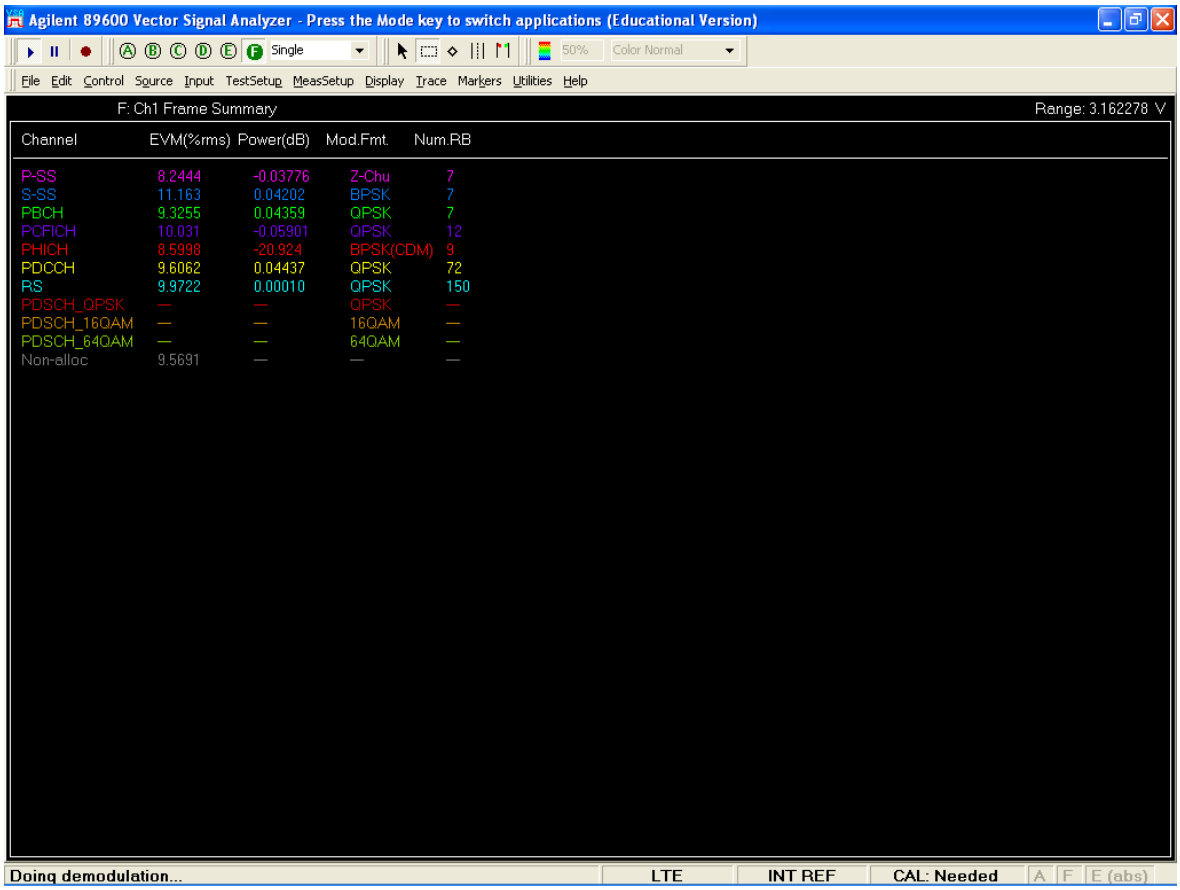



Figura 7.22: Valores del EVM para los distintos canales.

En la figura 7.22 tenemos expandida la ventana resumen con todos los canales que es capaz de detectar este paquete para la demodulación de señales LTE, así como una serie de parámetros que explicaremos a continuación (véase el capítulo 4). Es capaz de demodular las señales de sincronización P-SS y S-SS, los canales PBCH, *Physical Control Format Indicator Channel* (PCFICH), PHICH, PDCCH, *Physical Downlink Shared Channel* (PDSCH), y una señal de referencia RS. Admás, nos da el EVM, la potencia de recepción de cada uno de los canales anteriormente mencionados, la modulación con la que se transmiten y el número de radio bloques detectado. El parámetro que más nos interesa es el EVM, que según la especificación [1] debe ser inferior al 17,5 % para la modulación QPSK, menor que 12,5 % para 16-QAM, y menor que 8 % para 64-QAM. En este caso los valores obtenidos para cada uno de los canales son inferiores a los estipulados.

### *Power Spectral Density* y espectrograma mediante MATLAB

En este apartado se ha realizado la captura de la señal LTE de nuestra estación base utilizando Simulink utilizando el paquete *espectro_lte2.slx* para posteriormente procesarla con un paquete de MATLAB llamado *LTE System Toolbox*$^{TM}$. El paquete *LTE System Toolbox*$^{TM}$ puede utilizarse para sincronizar, demodular, y decodificar una señal de un eNB en tiempo real.

La estructura del paquete de *Simulink* que hemos utilizado (*espectro_LTE*) se puede visualizar en la figura 7.23. Con ella hemos obtenido en tiempo real la PSD y el espectro de la señal transmitida por el OAI eNB.

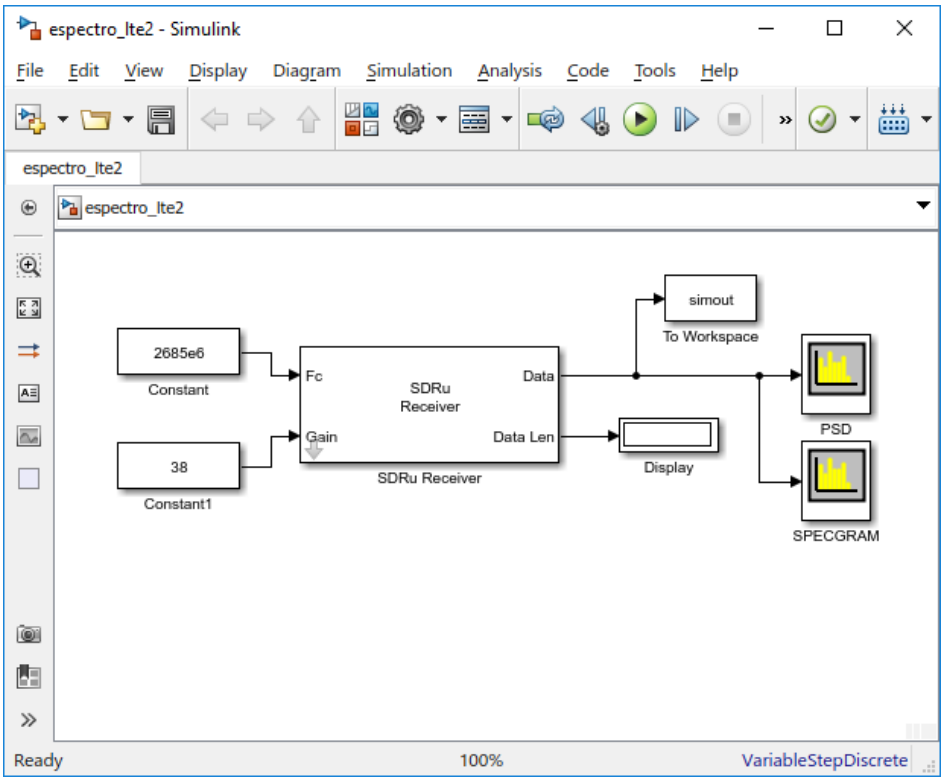



Figura 7.23: Paquete *espectro lte2* de Simulink.

En primer lugar, una vez realizada la conexión de algún UE a nuestra red LTE, se ha detectado la señalización entre el eNB y el UE, en el enlace ascendente.

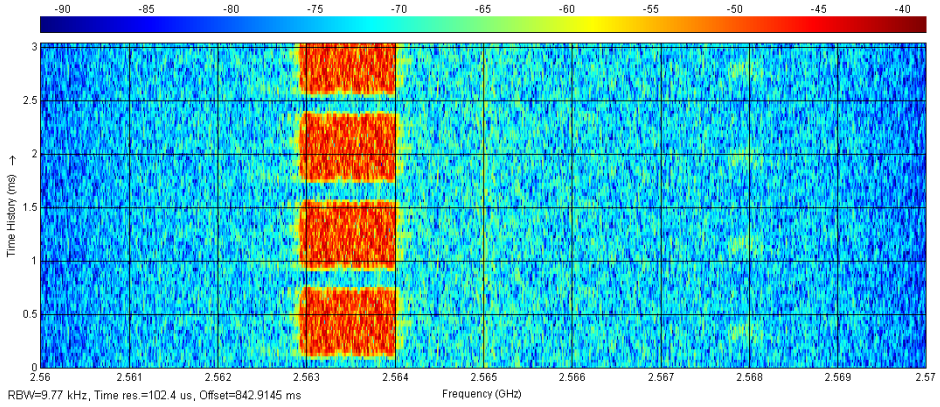



Figura 7.24: Espectrograma de señalización en UL.

La figura 7.24 es el espectrograma una vez que se ha conectado el UE a la red móvil, y se produce el intercambio de información entre el UE y el OAI eNB. Observamos que únicamente se utilizan frecuencias inferiores de todas las disponibles en el espectro. El ancho de banda utilizado es aproximadamente de 1 MHz, lo que equivale a 5 grupos de 12 subportadoras por grupo, teniendo en cuenta la separación entre ellas: $\triangle f = 15kHz$. También, podemos observar diferentes *slots* temporales con la duración establecida por el estándar de 0.5 ms. Estas ráfagas de información pueden ser a causa de actualización de información de la red, solicitudes por parte de la red al UE, o el caso contrario, actualizaciones de aplicaciones por parte del UE, etc.

En segundo lugar, accederemos a una página web mediante el UE, para poder capturar el intercambio de información en el enlace ascendente, obteniendo así, el espectrograma y la PSD en la figura 7.25.

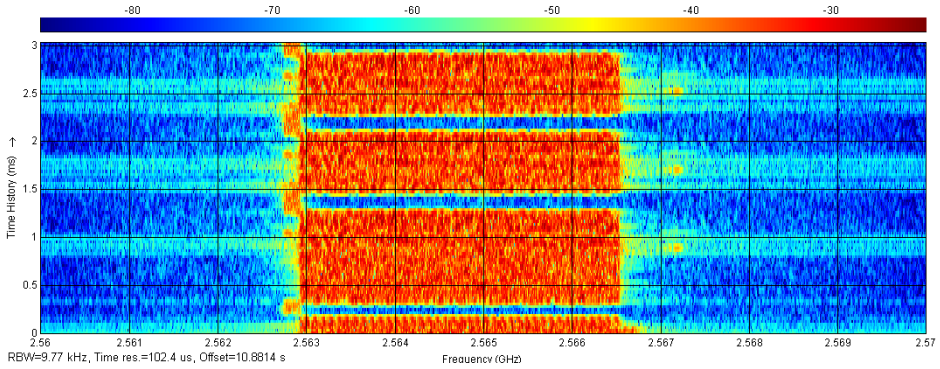



Figura 7.25: Espectrograma del intercambio de información en UL.

En la figura 7.25 observamos la ocupación del ancho de banda en el canal ascendente, debido a las diferentes peticiones que se producen entre la red móvil y el UE. Del mismo modo podemos comprobar que solamente se utilizan 3,5 MHz de los 4,5 MHz disponibles para el ancho de banda de 5 MHz. Esto puede deberse a que el UE utilizado es incapaz de soportar todo el ancho de banda, o por limitaciones de OpenAirInterface en el sentido ascendente. Además, en la parte superior del espectrograma podemos observar los *slots* de 0.5 ms, mientras que en la parte inferior se ha capturado una subtrama de 1 ms. Ésta es una de las 10 subtramas que forman una trama de LTE. La separación temporal sabemos que está establecida en 0.21ms, que debe ser menor que el tiempo de coherencia del canal, por lo que también comprobamos que se ajusta correctamente a las especificaciones.

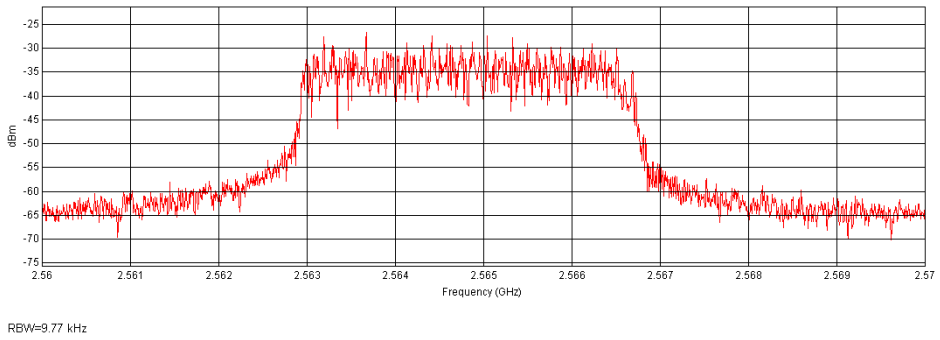



Figura 7.26: PSD del intercambio de información en UL.

Una vez más, confirmamos el resultado obtenido de la PSD por el analizador de espectros para el enlace ascendente en la figura 7.26. En ella observamos que únicamente utiliza un ancho de banda nominal de 3,5 MHz, siendo infrautilizadas las frecuencias superiores del enlace ascendente.

En tercer lugar, realizaremos el mismo procedimiento anterior pero esta vez para el enlace descendente. En la figura 7.27 tenemos el espectrograma del intercambio de información entre la red y el UE sin realizar ninguna prueba, únicamente para

comprobar qué utilización del canal se produce cuando no se realiza ninguna acción con el UE.

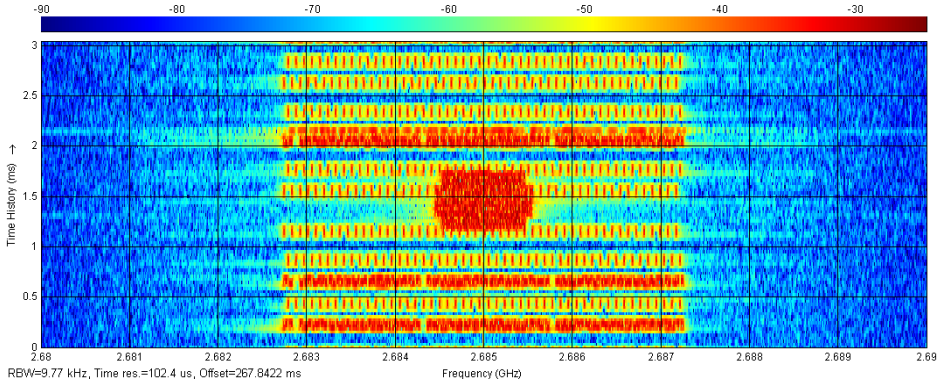



Figura 7.27: Espectrograma de señalización en DL.

En la figura anterior observamos una disminución de la utilización del canal DL, aunque se aprecian algunas ráfagas debido a la señalización de la propia red móvil, como la necesaria para la sincronización temporal, identificación de la célula o la estimación del canal. Comprobamos que en esta ocasión, sí se encuentra disponible todo el ancho de banda nominal que ascendería a los 4,5 MHz para los 5 MHz como ancho de banda total. Además, podemos observar la transmisión de información de las distintas subportadoras en distintos *slots* temporales, así como los grupos de 12 subportadoras.

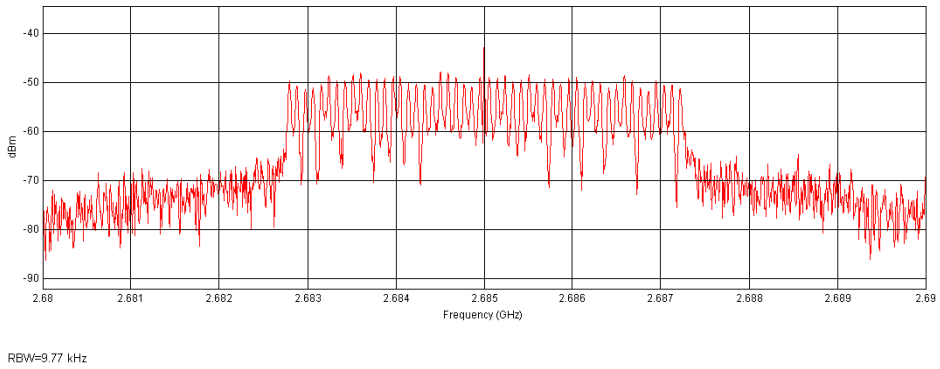



Figura 7.28: PSD de señalización en DL.

Por otro lado, en la figura 7.28 nos encontramos con la PSD de señalización, en la que corroboramos el espectrograma anterior. Solamente se utilizan unas determinadas subportadoras, ya que en esta situación no es necesaria la utilización de todo el ancho de banda disponible.

Finalmente, mediante el UE realizaremos una conexión a Internet a través de la red LTE, descargando un recurso de gran tamaño para analizar el comportamiento del enlace descendente, como puede ser la descarga de un fichero de texto, o la visualización de un vídeo online.

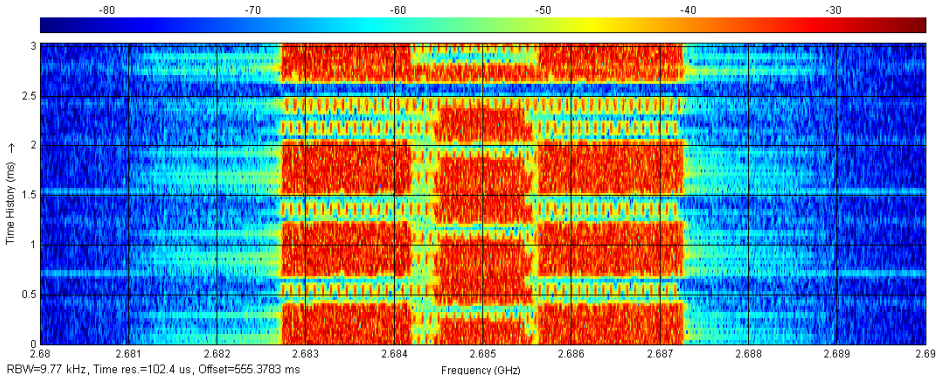



Figura 7.29: Espectrograma en DL.

En la figura 7.29 observamos un aumento de la utilización del canal debido al envío de información al UE. Aunque vemos un aumento significativo de *slots* temporales y activación de subportadoras, no estamos utilizando todo el ancho de banda debido a que las solicitudes de información no son suficientemente grandes. Existen intervalos en los que apenas no se produce transmisión de información hacia el UE. Además, se observa la transmisión de información en distintos *slots* temporales, así como en distintas agrupaciones de subportadoras.

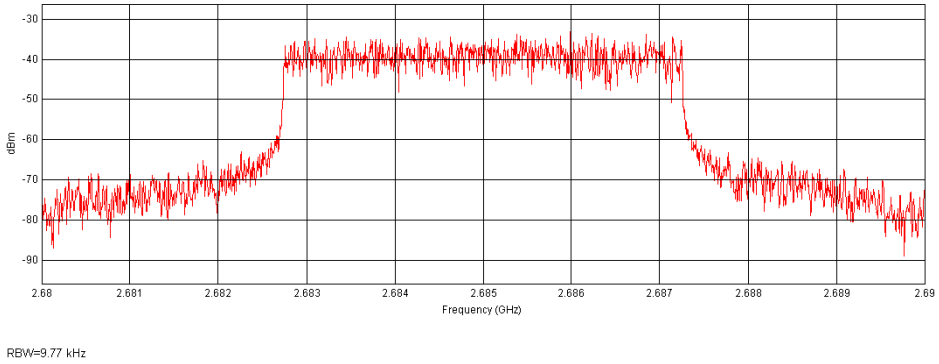



Figura 7.30: PSD en DL.

En la figura 7.30 volvemos a reafirmar el aumento de la utilización del canal descendente. De nuevo corroboramos los resultados obtenidos por el analizador de espectros, afirmando que en el enlace descendente tiene un ancho de banda nominal de 4,5 MHz.

### Funcionamiento de *LTE System Toolbox™*

Una vez realizadas la captura de las señales de nuestra estación base con distintos ancho de banda, y almacenadas en los ficheros .mat creados por el paquete de Simulink *"espectro_lte.slx"*, procederemos a su análisis mediante *LTE System Toolbox™*.

Para que dicho paquete pueda comunicarse con la red, primero se deben realizar procedimientos de búsqueda y selección de celdas, así como obtener información de la red. Estos procedimientos conllevan a la sincronización de ranuras y tramas, descubrir el identificador de celda, y decodificar los bloques *Master Information Block* (MIB) y SIB1. El bloque MIB contiene la información del ancho de banda utilizado, el *System Frame Number* (SFN) y la configuración del canal PHICH.

El MIB se transmite en el canal de difusión BCH, asignado al canal físico PBCH, transmitiéndose un esquema fijo de codificación y modulación. Con la información obtenida del MIB, el paquete de MATLAB o un UE puede decodificar el CFI, que indica la longitud el canal PDCCH, permitiendo que éste se decodifique y busque mensajes de información del control del enlace descendente (DCI). Finalmente el formato y la asignación de recursos de transmisión PDSCH está indicada por un mensaje DCI en el canal físico PDCCH. Este es el procedimiento que utiliza éste paquete, para realizar la decodificación, demodulación y la obtención de información de la red.

Por otro lado, sabemos que el MIB corresponde a un bloque de transporte BCH, y el tiempo necesario para transmitir un bloque son 40 ms o 4 tramas. El BCH se transmite en 4 partes, y cada una de ellas se asigna a la subtrama 0 de una trama. Para asegurar que se recibe la subtrama 0 debemos capturar al menos 11 subtramas para tener en cuenta que la captura se inicie durante la primera subtrama (cada trama en LTE se compone de 10 tramas de 1 ms cada una). Para poder capturar las 4 partes del MIB, necesitaremos al menos 41 tramas. Lo mismo ocurre con el SIB, el cual se transmite en la subtrama 5 de cada trama par, y además cuenta con 4 versiones diferentes de redundancia; lo que se transmite consecutivamente cada 80 ms u 8 tramas. Por lo que necesitamos capturar al menos 21 subtramas para asegurar la recepción de una versión, ó de 81 subtramas para capturar todas las versiones.

Antes de decodificar el MIB, no conocemos el ancho de banda del sistema, por lo que las señales de sincronización primaria (PSS) y secundaria (SSS) y el PBCH (que contiene el MIB), se encuentran en las 62 subportadoras centrales (aunque se reservan 72 subportadoras). De esta forma el paquete *LTE System Toolbox*$^{TM}$ demodula inicialmente está región.

Para la búsqueda de celdas, la detección del prefijo cíclico y la detección del modo dúplex, este paquete realiza una llamada a la función *lteCellSearch*. La búsqueda se repite para cada combinación de la longitud del prefijo cíclico y del modo dúplex, realizándose la correlación entre ellos y obteniendo el máximo, permitiendo identificar dichos parámetros. Así obtenemos un gráfico de la correlación entre la señal recibida y el PSS/SSS para identificar la celda detectada.

Para la demodulación OFDM y la estimación del canal, la señal se desmodula produciendo una cuadrícula que se utiliza para realizar la estimación del canal. Tenemos los parámetros *"hest"* para la estimación del canal, *"nest"* para la estimación del ruido, y *"cec"* que es la configuración del estimador del canal. La estimación del canal se realiza solamente en la primera subtrama, utilizando por ejemplo los primeros 1000 símbolos OFDM. Se utiliza una ventana de 9x9 en la escala del tiempo y en frecuencia para reducir el impacto del ruido en las estimaciones piloto durante dicha estimación.

Demodulación PBCH, decodificación BCH y MIB. El MIB se decodifica extrayendo los RE correspondientes al canal PBCH desde la primera subtrama a través de todas las antenas de recepción y la estimación del canal. La función *"ltePBCHDecode"* establece la temporización de trama en módulo 4 y devuelve el resultado en el parámetro *"nfmod4"*. Además, alamacena los bits de información del MIB en un vector para sus posterior análisis. La función *"lteMIB"* se utiliza para analizar los bits del vector MIB y añadir la información relevante a la estructura de configuración *"enb"*.

Demodulación OFDM en todo el ancho de banda. Una vez que se conoce el ancho de banda de la señal, se vuelve a realizar su muestreo a la frecuencia nominal utilizada para ese ancho de banda. La estimación y la corrección del desplazamiento de frecuencia se realiza en la señal muestreada, llevándose a cabo la sincronización y la demodulación OFDM.

En primer lugar, realizaremos el análisis de la señal capturada a partir de Simulink, cuando nuestro eNodeB se encontraba transmitiendo con 25 bloques de recursos (RBs), es decir, con un ancho de banda de 5 MHz.

Una de las imágenes que nos muestra el paquete LTETolbox de MATLAB es la respuesta en magnitud del canal. Como vemos en la figura 7.31, para un ancho de banda de 5MHz, LTE establece un total del 301 portadoras. Además, en esta imagen podemos observar la magnitud en el tiempo de 15 símbolos OFDM.

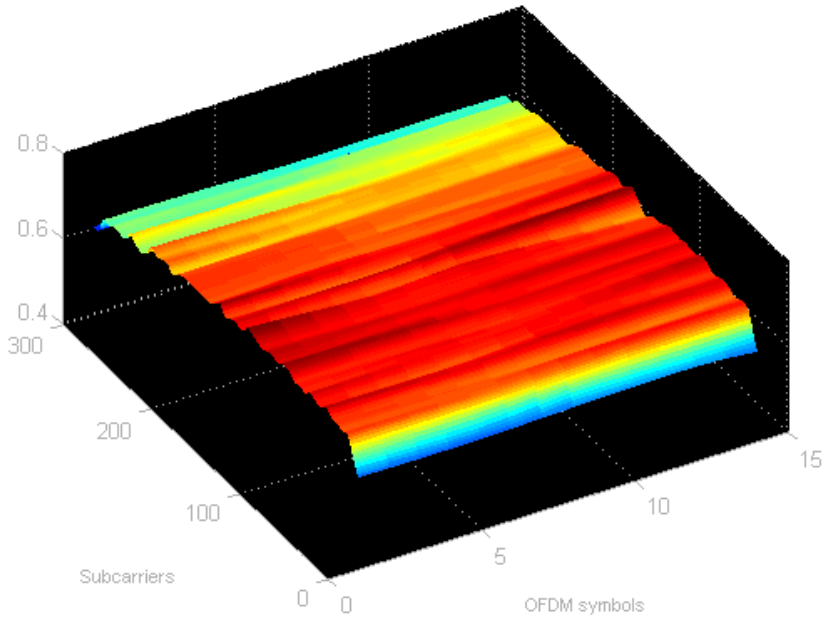



Figura 7.31: Respuesta en magnitud del canal para un BW de 5MHz.

En la figura 7.32, recoge el espectro obtenido. En este podemos ver el máximo, mínimo y el valor medio del espectrograma de la señal. Vemos que oscila en el rango comprendido entre -30 dBm y -80 dBm.

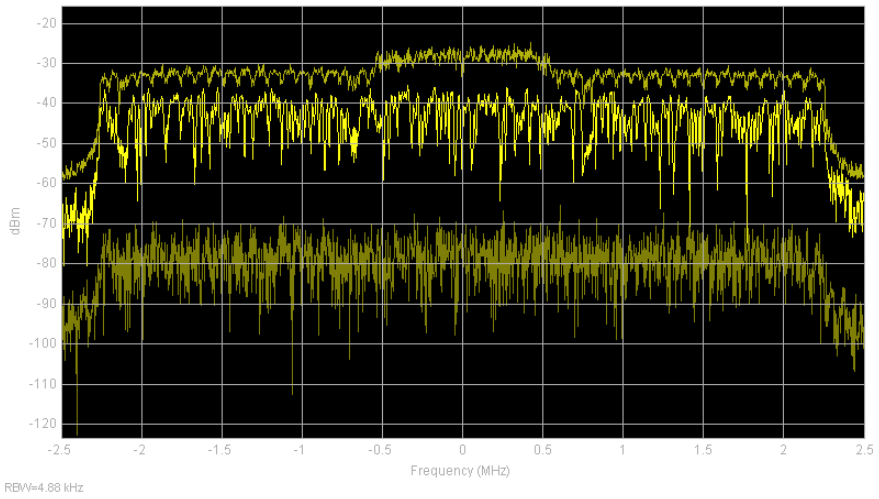



Figura 7.32: Espectro del canal para un BW de 5MHz.

En la figura 7.33, nos muestra la constelación QPSK para el canal decodificado correspondiente al canal PDSCH. Para dicho canal se han estimado los siguientes valores RMS EVM: 6.352 %, PDSCH Peak EVM: 15.591 %

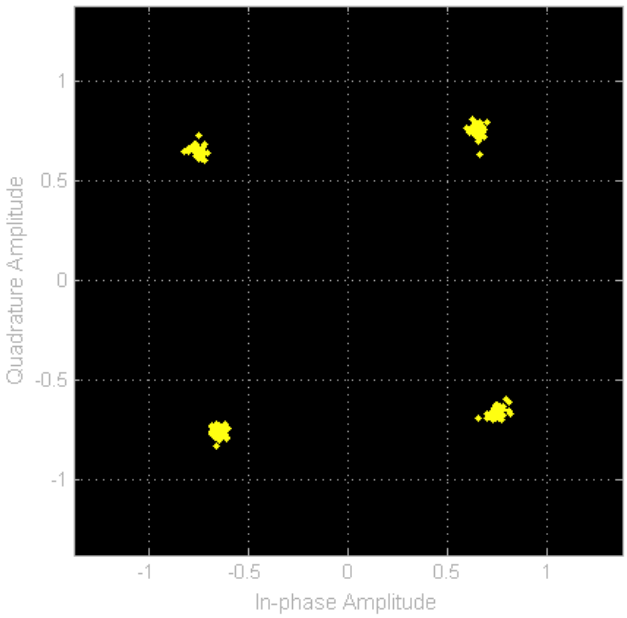



Figura 7.33: Constelación del canal para un BW de 5MHz.

La figura 7.34, se corresponde con la correlación para identificar el principio de las tramas. Se correla la señal recibida con la señal de sincronización de trama (PSS, que es una secuencia de bits). Es necesario la sincronización de trama antes que la demodulación de los bits. A partir de los bits de la PSS se generan los símbolos OFDM, y esa es la señal que se correla con la señal recibida (aún no está demodulada, no se puede correlar a nivel de bits).

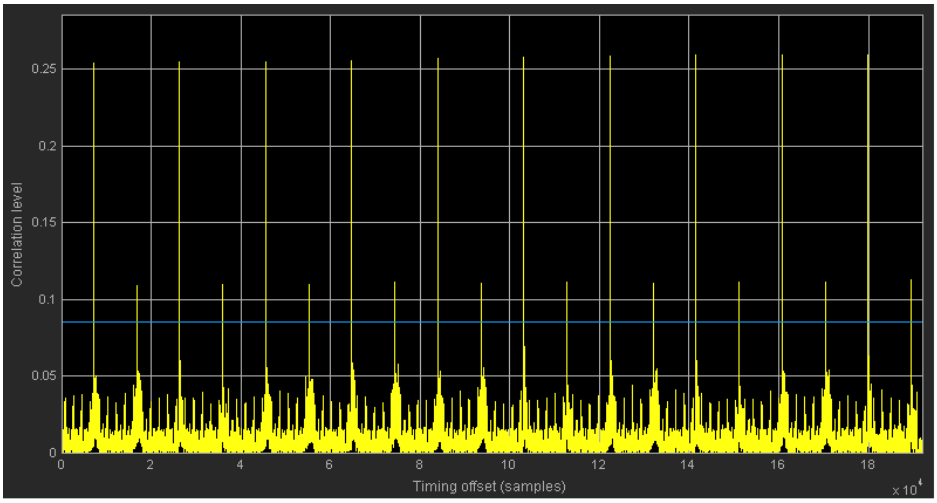



Figura 7.34: Correlación del canal para un BW de 5MHz.

De nuevo en la figura 7.35, tenemos la evolución en el tiempo y en frecuencia del canal, pero esta vez para una configuración de 50 RB, que hace un total de 10 MHz de

ancho de banda, y un total de 601 subportadoras utilizadas.

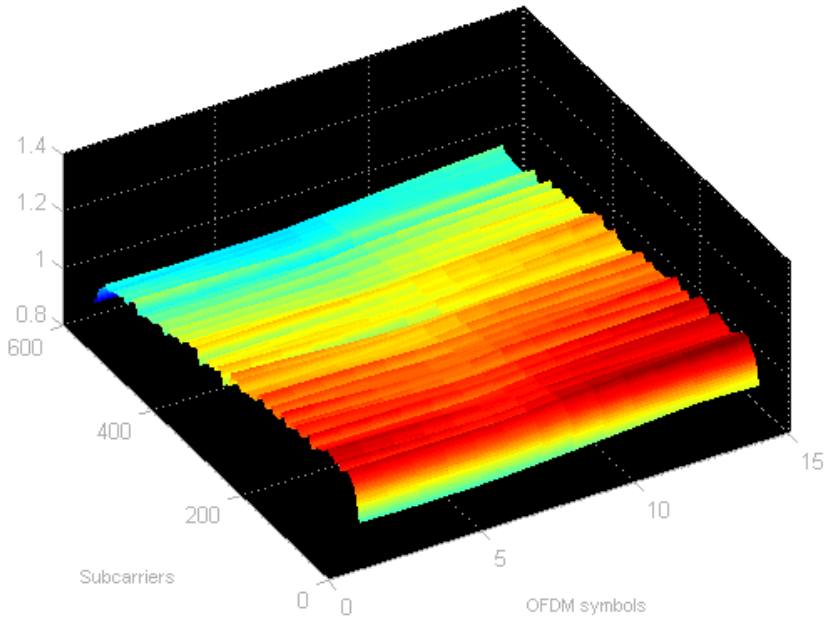



Figura 7.35: Respuesta en magnitud del canal para un BW de 10MHz.

En la figura 7.36, observamos el espectrograma para esta nueva configuración de 10 MHz. En esta imagen observamos que el mínimo y el valor medio han reducido su valor respecto a la configuración anterior. En este lugar, la potencia oscila en el rango de -30 dBm y -90 dBm, mientras que los valores medios para cada una de las subportadoras se sitúa en torno a los -50 dBm.

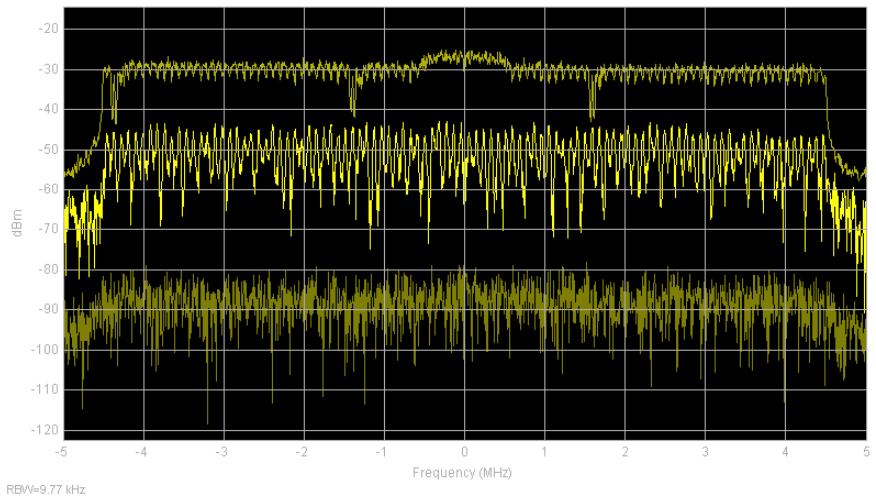



Figura 7.36: Espectro del canal para un BW de 10MHz.

De nuevo en la figura 7.37, observamos la constelación QPSK para el canal PDSCH, pero esta vez hemos obtenido como resultado los valores: RMS EVM: 10.663 % Peak EVM: 25.225 %

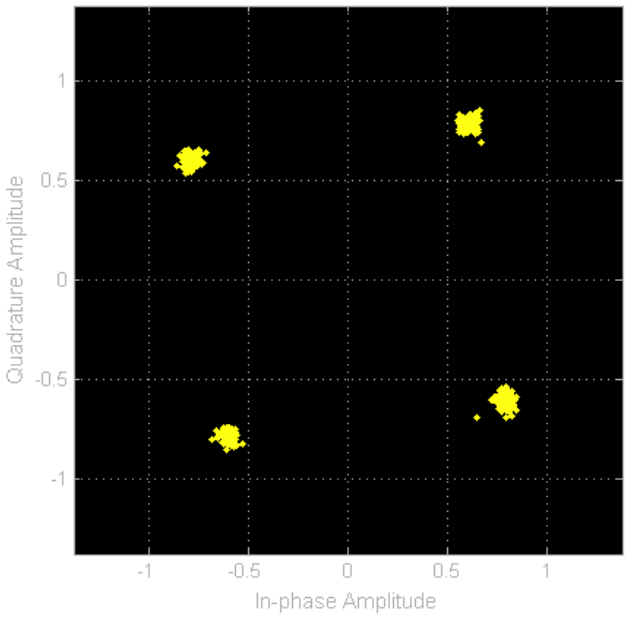



Figura 7.37: Constelación del canal para un BW de 10MHz.

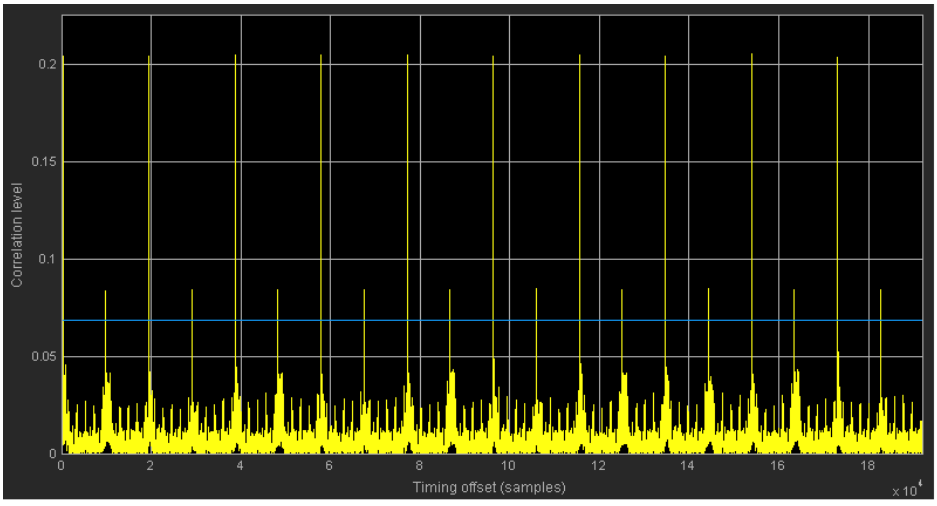



Figura 7.38: Correlación del canal para un BW de 10MHz.

En la figura 7.39, esta vez observamos la respuesta del canal en magnitud para una configuración de ancho de banda de 20 MHz (100 RB), lo que supone una utilización de 1001 subportadoras.

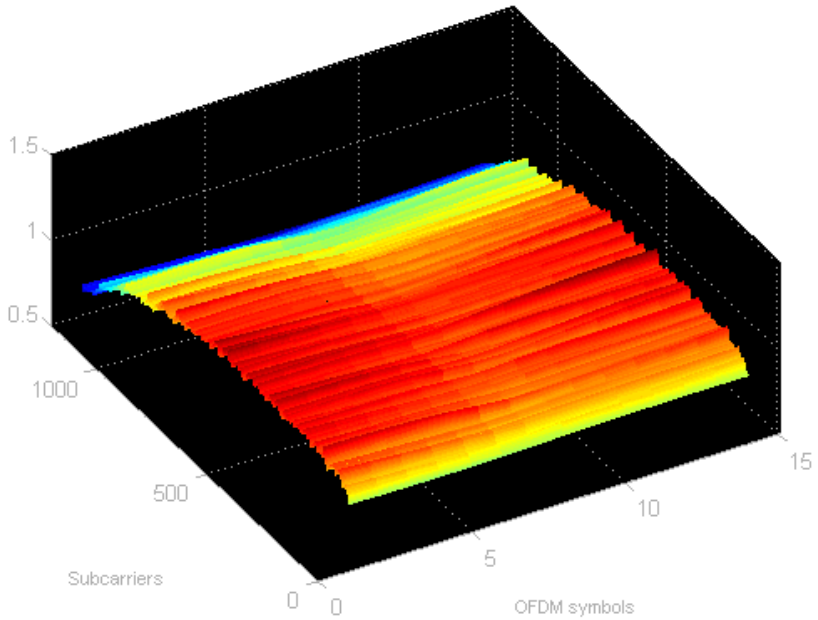



Figura 7.39: Respuesta en magnitud del canal para un BW de 20MHz.

Respecto, al espectro obtenido para esta configuración, lo observamos en la figura 7.40. Obteniendo, en torno a los -30dBm para los valores máximos, -90 dBm para los mínimos, y situándose la media en torno a los -60 dBm.

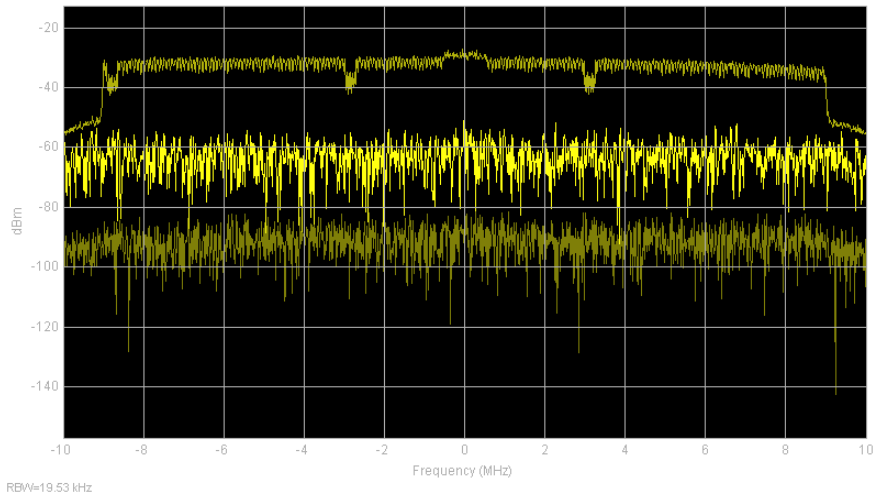



Figura 7.40: Espectro del canal para un BW de 20MHz.

En la figura 7.41, se ha obtenido la constelación QPSK respecto del canal PDSCH. Obteniendo como resultado los siguientes valores según la información obtenida del SIB1: PDSCH RMS EVM: 16.750 %, PDSCH Peak EVM: 41.406 %

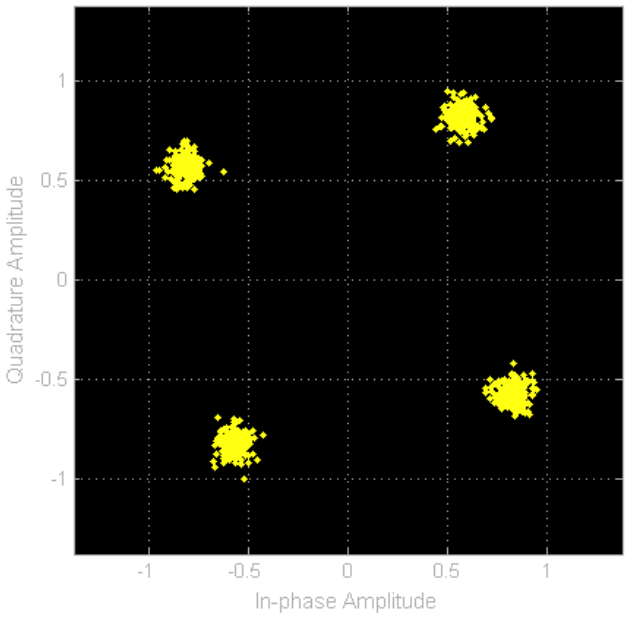



Figura 7.41: Constelación del canal para un BW de 20MHz.

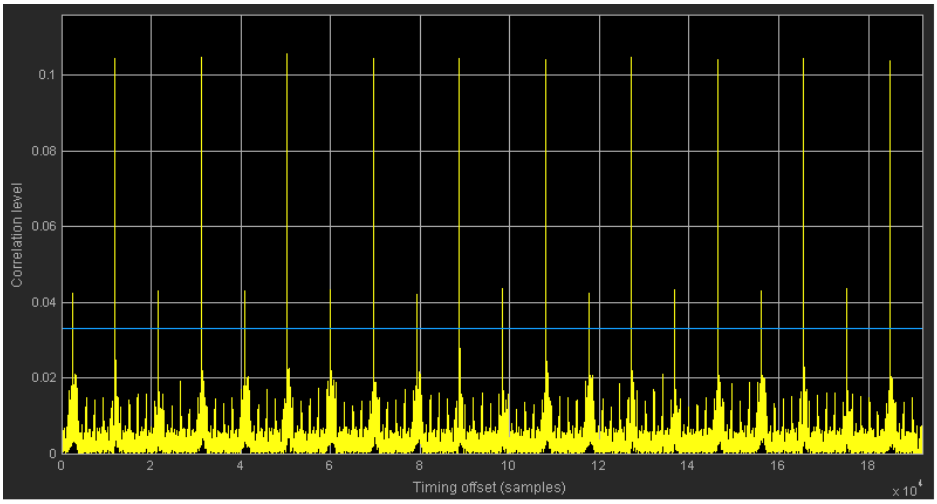



Figura 7.42: Correlación del canal para un BW de 20MHz.

### 7.3.2. Análisis de trazas

En este apartado, realizaremos el análisis de la conexión a la red del UE a partir de las trazas obtenidas con *Wireshark*. Para realizar dicha prueba, hemos utilizado los dos móviles disponibles. En el móvil Xiaomi Mi5 se ha insertado la USIM número 12, y en el BQ X5 la USIM 11. Una vez que tenemos funcionando el eNB y el EPC de OAI, y previamente configurados en modo avión para evaluar correctamente que no están conectados a la red móvil, lanzaremos *Wireshark* y desactivaremos el modo avión en el que se encuentran los dispositivos, para que se conecten a nuestra red móvil.

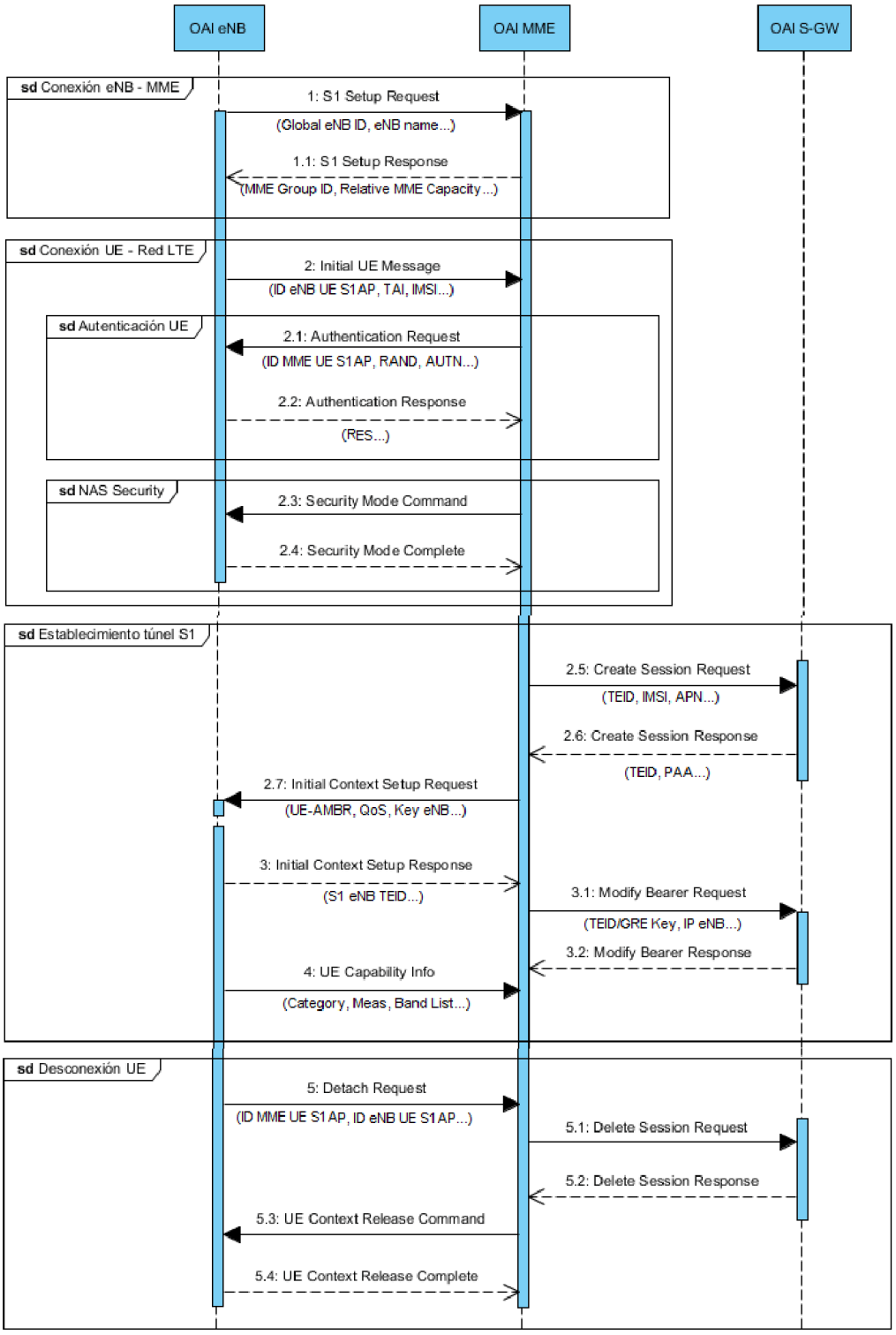



Figura 7.43: Flujo de mensajes entre las entidades eNB - MME - S-GW.

En la prueba de conexión a la red LTE con el dispositivo móvil Xiaomi MI5, observamos en la figura 7.44 que con este dispositivo se introducen dos mensajes más (*Identity Request, Identity Response*) debido a que en el mensaje *Initial UE Message* no se incluye el valor del IMSI. Por esto, la entidad MME solicita dicho parámetro al eNB con estos mensajes de solicitud y respuesta.

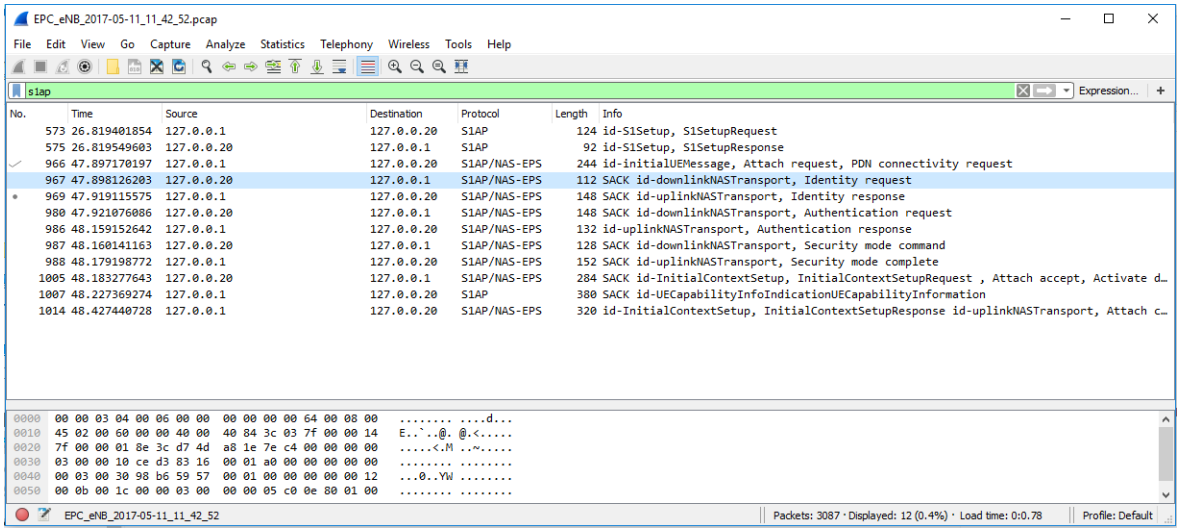



Figura 7.44: Trazas del UE Xiaomi MI5 del protocolo S1-AP

En la figura 7.45, vemos las trazas correspondientes con el protocolo S1-AP pertenecientes al dispositivo móvil BQ X5 Plus. Dicho protocolo, como vimos en el capítulo 4, es utilizado para dar servicios de señalización entre un eNB y el MME, y que éstos puedan interoperar correctamente en la interfaz S1.

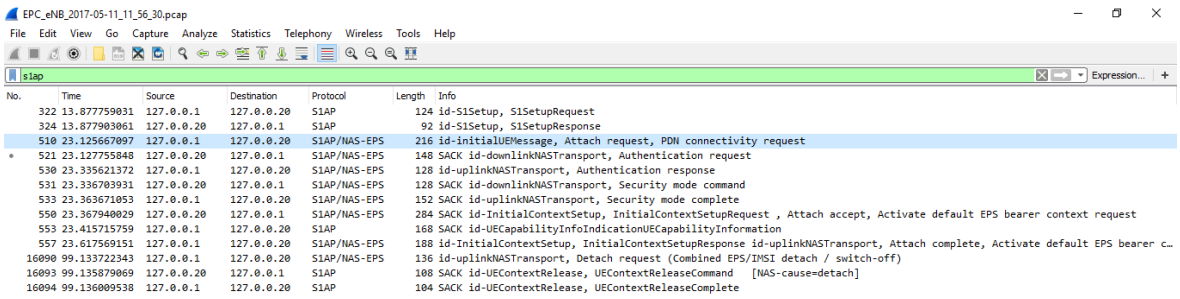



Figura 7.45: Trazas del UE BQ X5 Plus del protocolo S1-AP.

## Conexión eNB - MME

El eNB inicia el procedimiento enviando un mensaje *S1 Setup Request* incluyendo los datos apropiados para su asociación al MME, como pueden ser la identificación global del eNB, el MCC, el MNC, la identificación del propio eNB, etc. Estos campos se pueden ver en la figura 7.46.

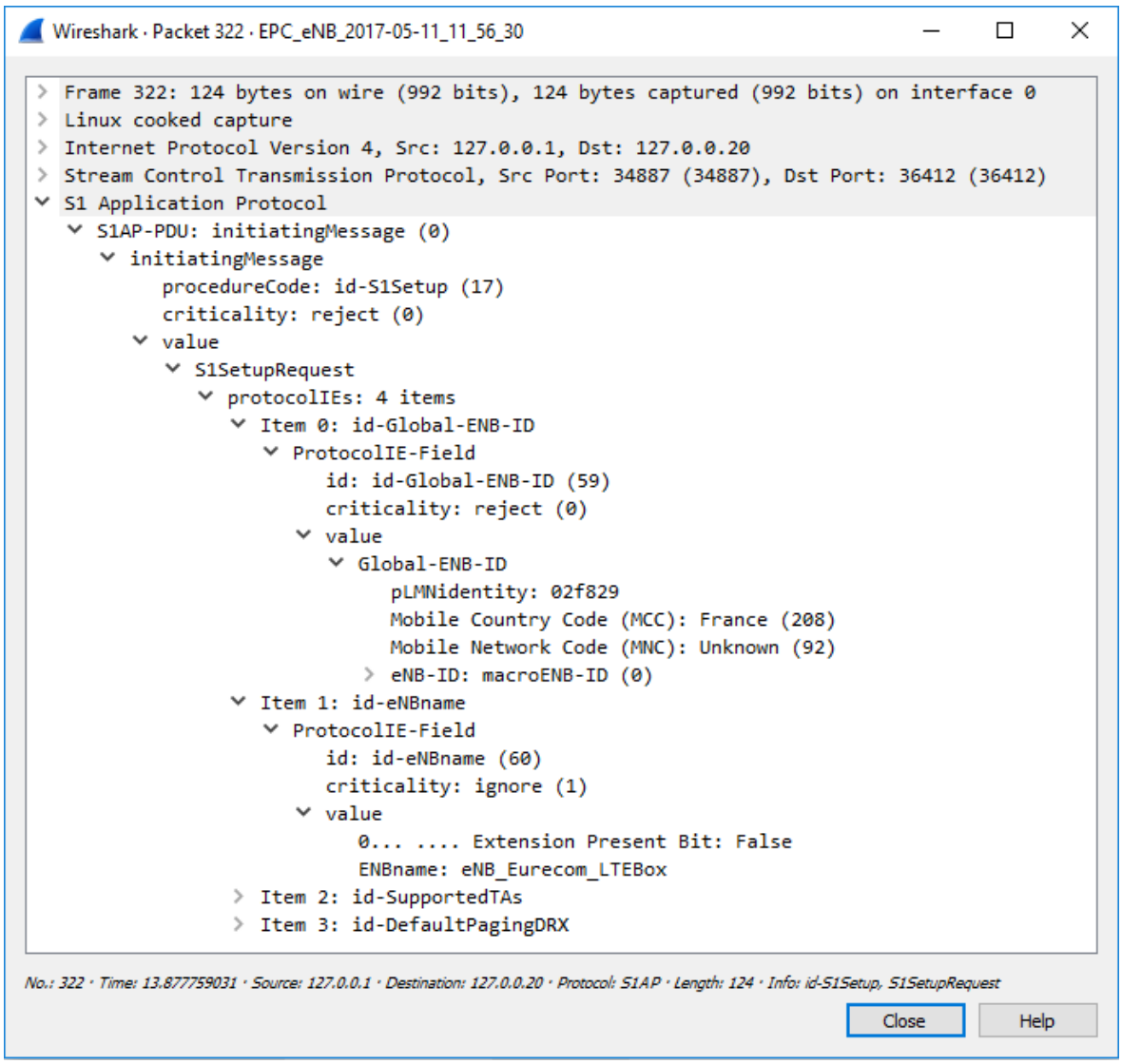



Figura 7.46: Campos del mensaje S1 Setup Request.

Acto seguido, el MME responde con el mensaje *S1 Setup Response* si no ha ocurrido ningún error en alguno de los datos verificando la información, o con el mensaje *S1 Setup Failure* en caso de error. Como podemos comprobar en la figura 7.45 se ha obtenido el mensaje de verificación. En este mensaje se envían de nuevo los datos como el MCC, el MNC, la identificación del grupo al que pertenece el MME, o el parámetro de capacidad relativa que se utiliza para realizar balanceo de carga entre MMEs. Estos parámetros son almacenados en el eNB, y éste ya podrá continuar con el intercambio de mensajes de la interfaz S1 (véase la figura 7.47).

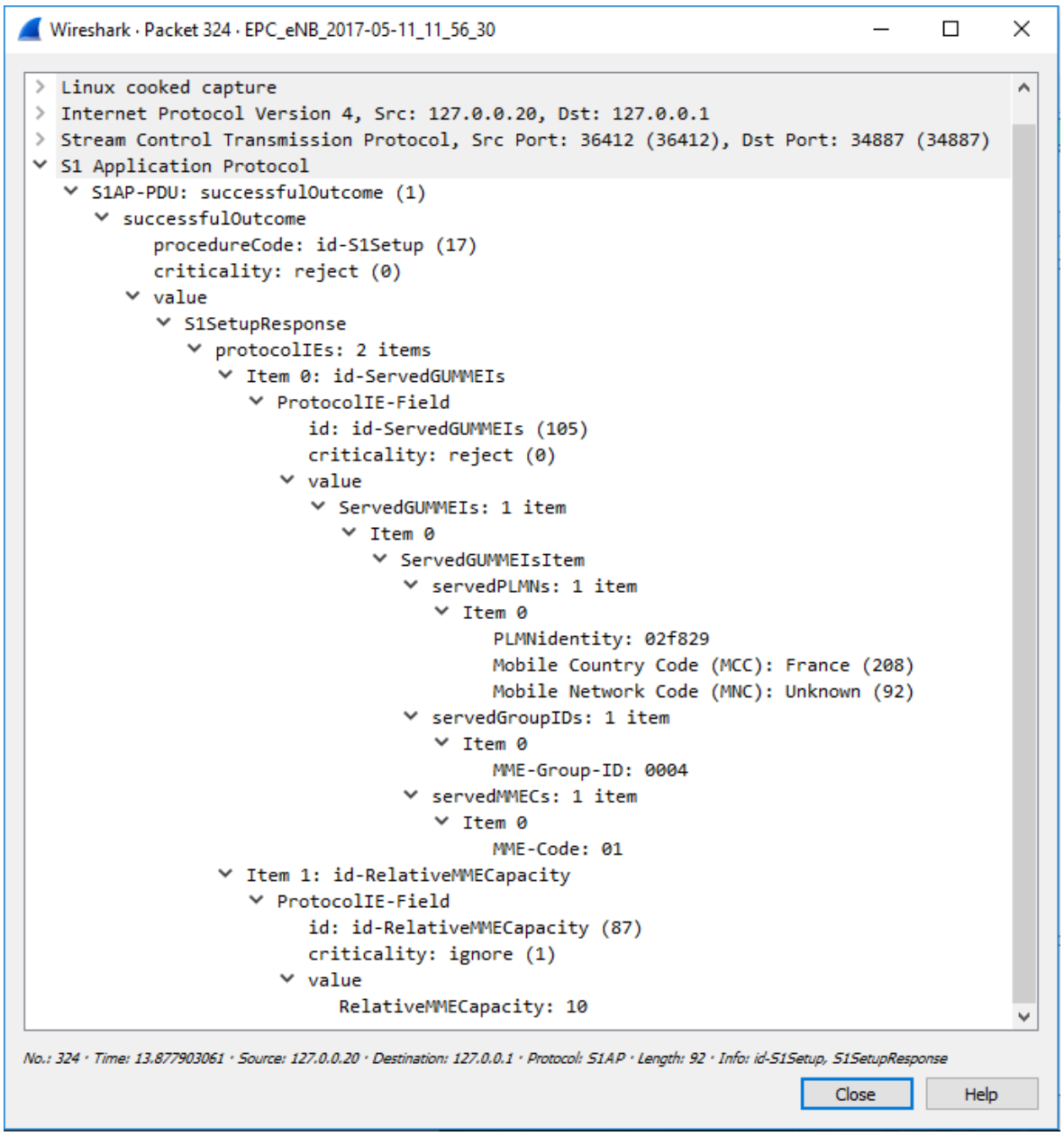



Figura 7.47: Campos del mensaje S1 Setup Response.

### Conexión UE - Red LTE

Una vez establecida la conexión entre el eNB y el MME, y analizado los mensajes intercambiados entre las dos entidades, realizaremos el análisis de los mensajes intercambiados cuando se conecta el móvil BQ X5 Plus a la red LTE. Como vimos en el capítulo 4, el propósito del protocolo NAS es llevar la señalización entre el UE y el MME a través de la interfaz S1.

El primer mensaje que se manda, si no existe ninguna conexión S1 asociada al UE en el eNB, es el *Initial UE Message*, como se puede apreciar en la figura 7.48. Dicho mensaje contiene 5 campos, los cuales contienen información sobre la conexión lógica S1-AP asociada por el UE - eNB, información del protocolo NAS, información sobre los códigos de la red móvil, e información sobre la conexión RRC.

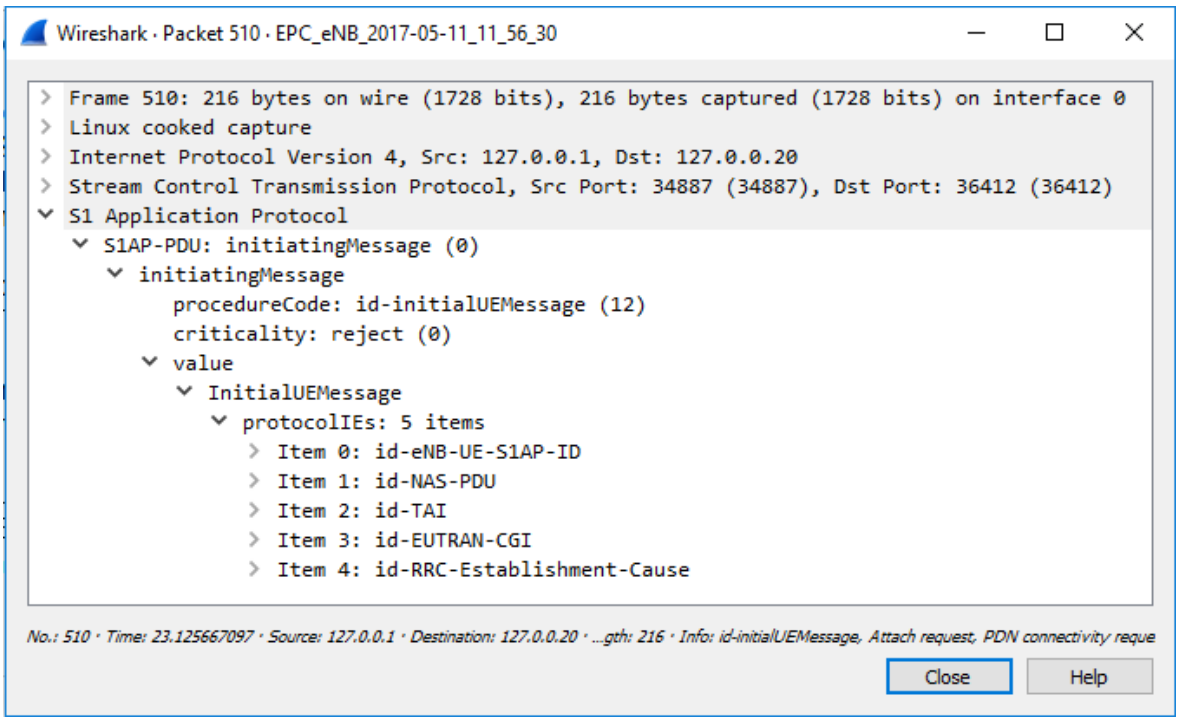



Figura 7.48: Campos del mensaje Initial UE message.

En dicho mensaje, el eNB le asigna un identificador único a esta conexión lógica, denominado *ENB-UE-S1AP-ID*. En nuestro caso, como podemos ver en la figura 7.49, se nos ha asignado el identificador 420141, se incluirán parámetros relacionados con el tipo de conexión, la identificación del UE (IMSI, MCC, MNC...), las capacidades del UE como algoritmos de encriptación, de integridad, etcétera. En el campo IMSI, corroboramos el valor del IMSI de la tarjeta USIM 11, introducida en el teléfono móvil BQ (véase la tabla 6.4).

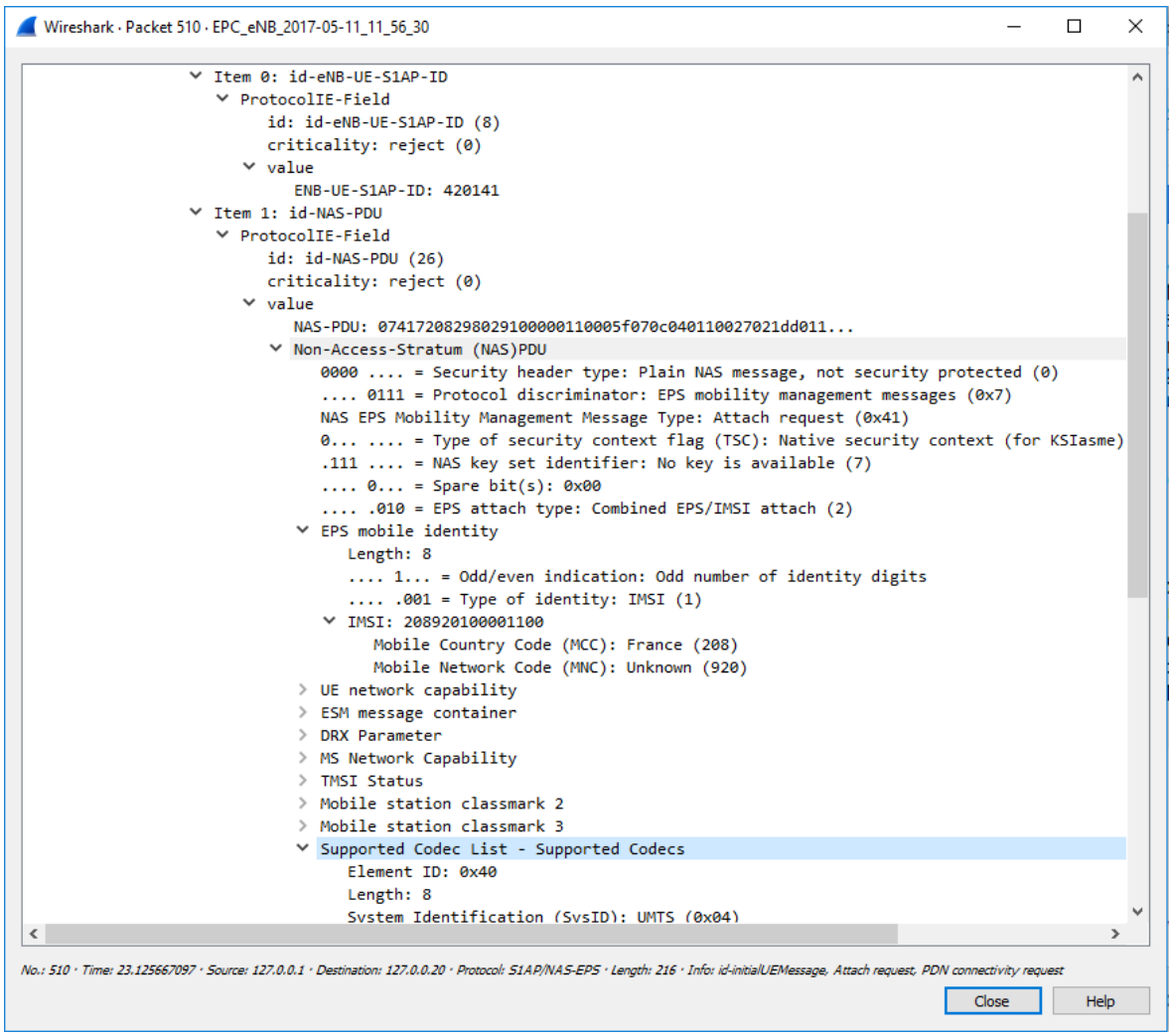



Figura 7.49: Valores de los distintos campos del mensaje Initial UE message.

Como destacamos al principio del apartado, la diferencia de los mensajes enviados depende del móvil usado. La figura 7.50 muestra el contenido del mensaje *Identity Response*. Podemos comprobar que el Xiaomi MI5 tiene insertada la tarjeta USIM 12 (véase la tabla 6.4).

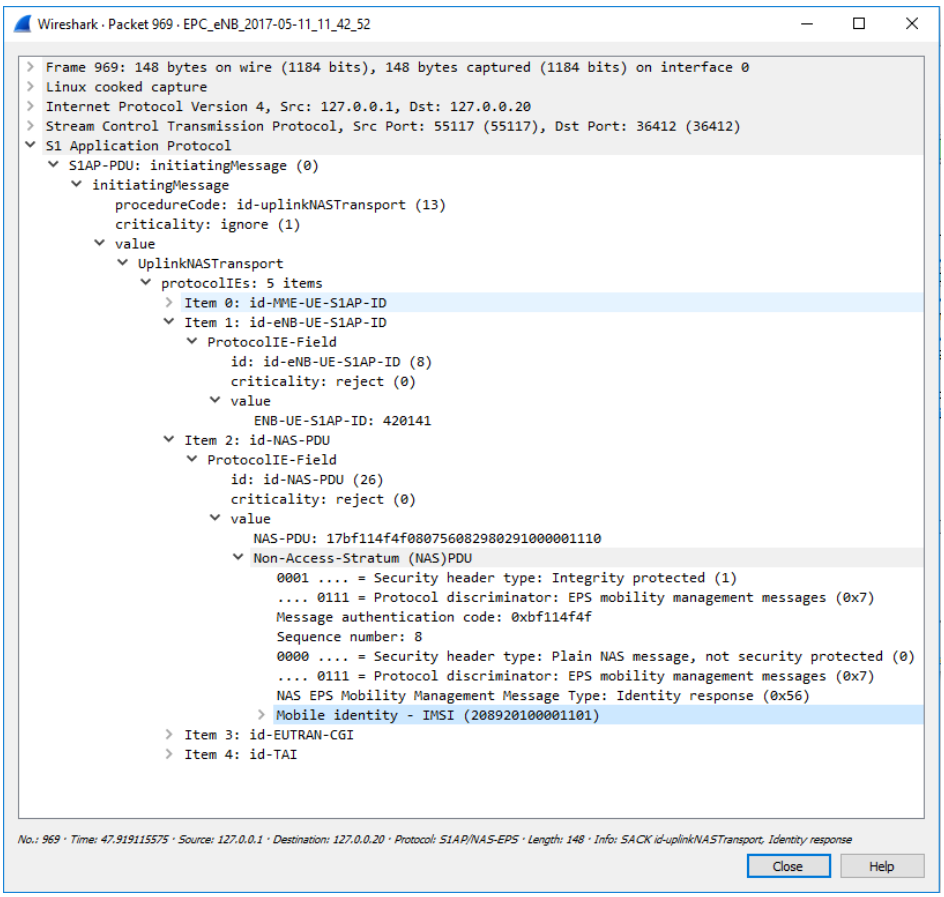
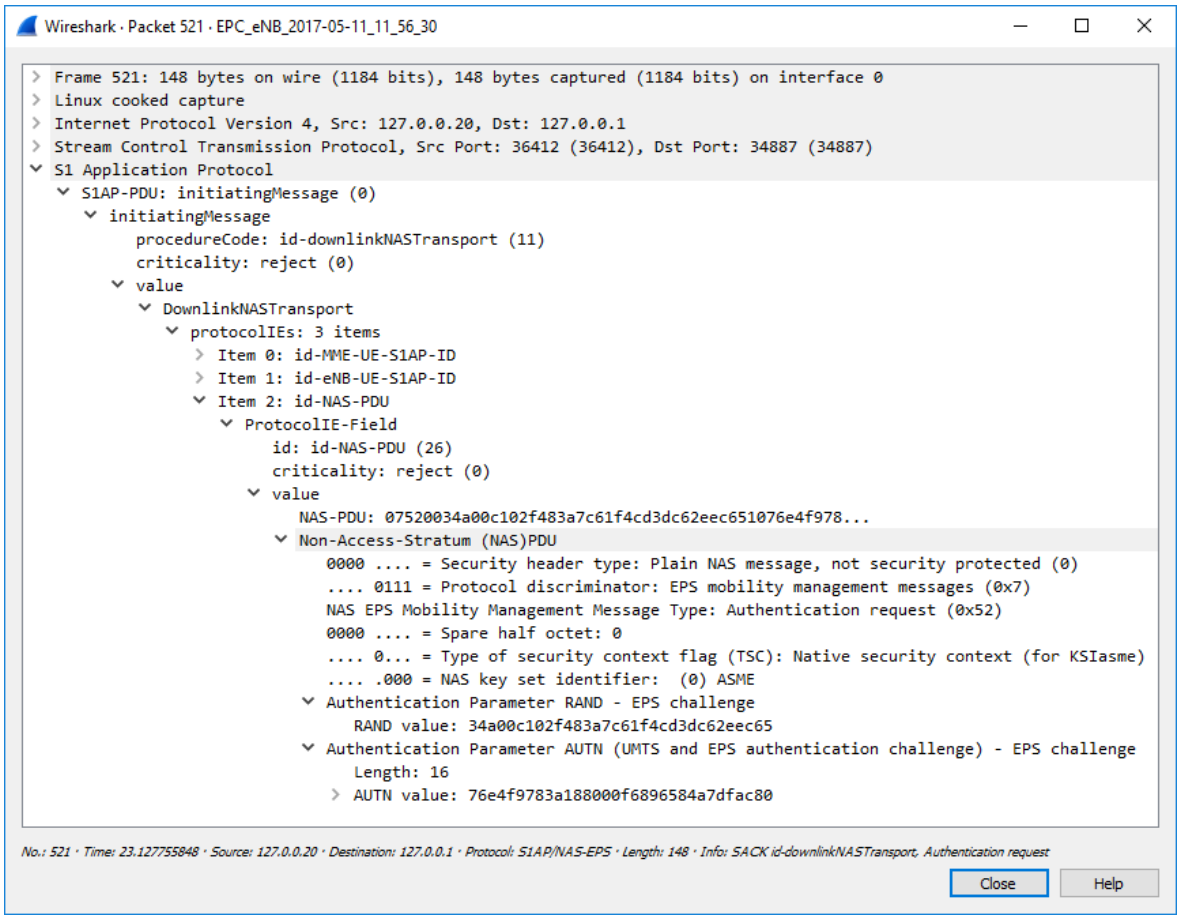
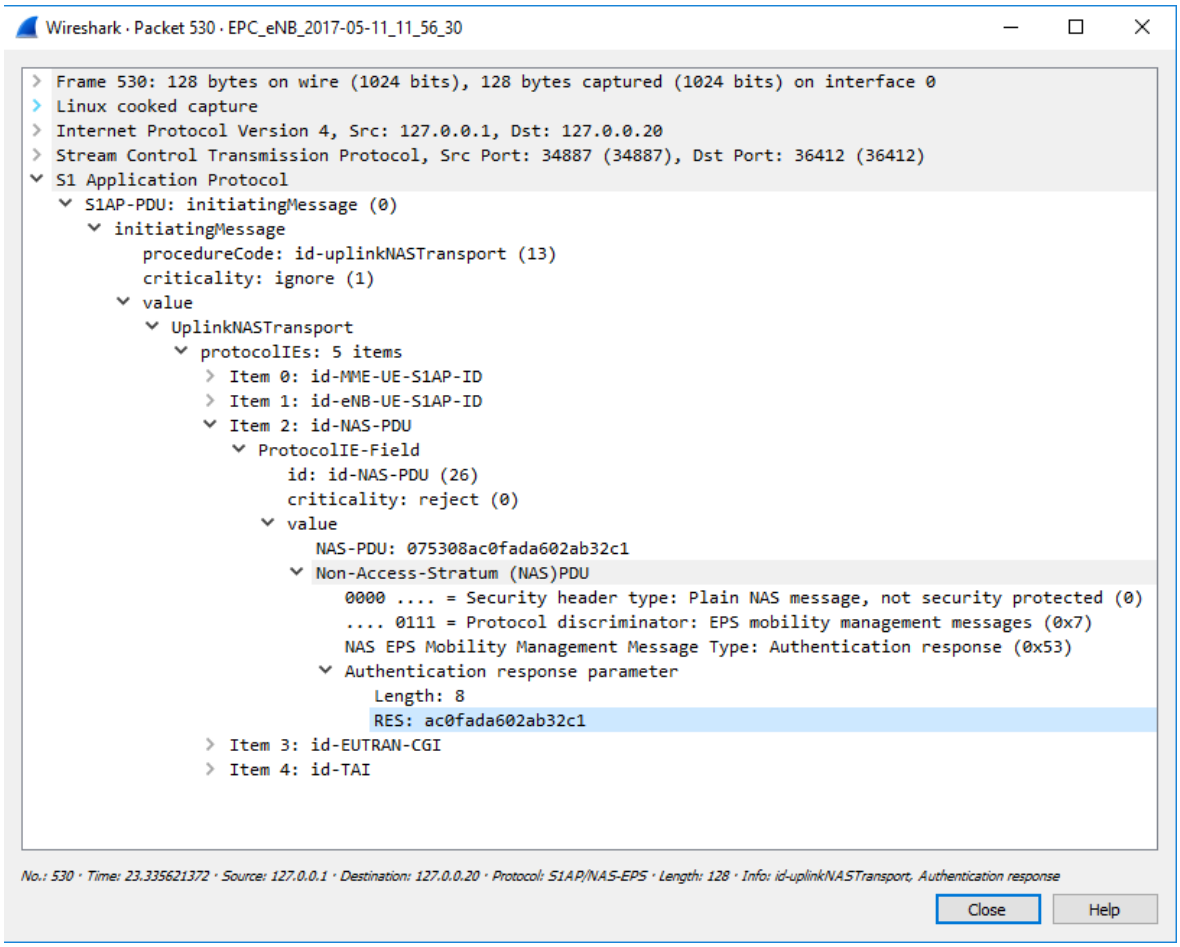



Figura 7.50: Mensaje *Identity Response* conteniendo el IMSI de la USIM 12.

El mensaje *Initial UE Message* es enviado al MME y, una vez recibido, éste le asigna un identificador *MME S1AP UE ID* a la conexión de señalización entre él y el eNB. El MME ejecuta una comprobación de integridad enviando el mensaje *Authentication Request*, en el que se incluyen los identificadores de la conexión lógica S1 establecida entre el UE - eNB, y el eNB - MME, y parámetros para la autenticación como el RAND, o el AUTN. Una vez que el UE recibe este mensaje, devolverá el resultado del desafío para la autenticación (parámetro RES) en el mensaje *Authentication Response*, que será comprobado por el MME (véase la figura 7.51).




(a) Authentication Request.




(b) Authentication Response.

Figura 7.51: Mensajes del procedimiento de autenticación.

Cuando la autenticación está completa, el UE y el MME generan las claves de seguridad NAS (KNA Senc, KNA Sint), dando lugar al procedimiento llamado *NAS Security Setup*. En éste se intercambian los mensajes *Security Mode Command* y *Security Mode Complete*.

Una vez finalizados los procedimientos de autenticación y de seguridad NAS, el MME envía un mensaje *Initial Context Setup Request* al eNB para que vaya estableciendo una conexión dedicada S1 con la entidad S-GW, y un *Data Radio Bearer* (DRB) con el UE. En él se incluyen, entre otros datos, las especificaciones para crear dicha conexión, como el *Bit Rate* en el enlace ascendente y descendente, la dirección IP del S-GW, el identificador de la QoS, o la dirección IP asignada al UE.

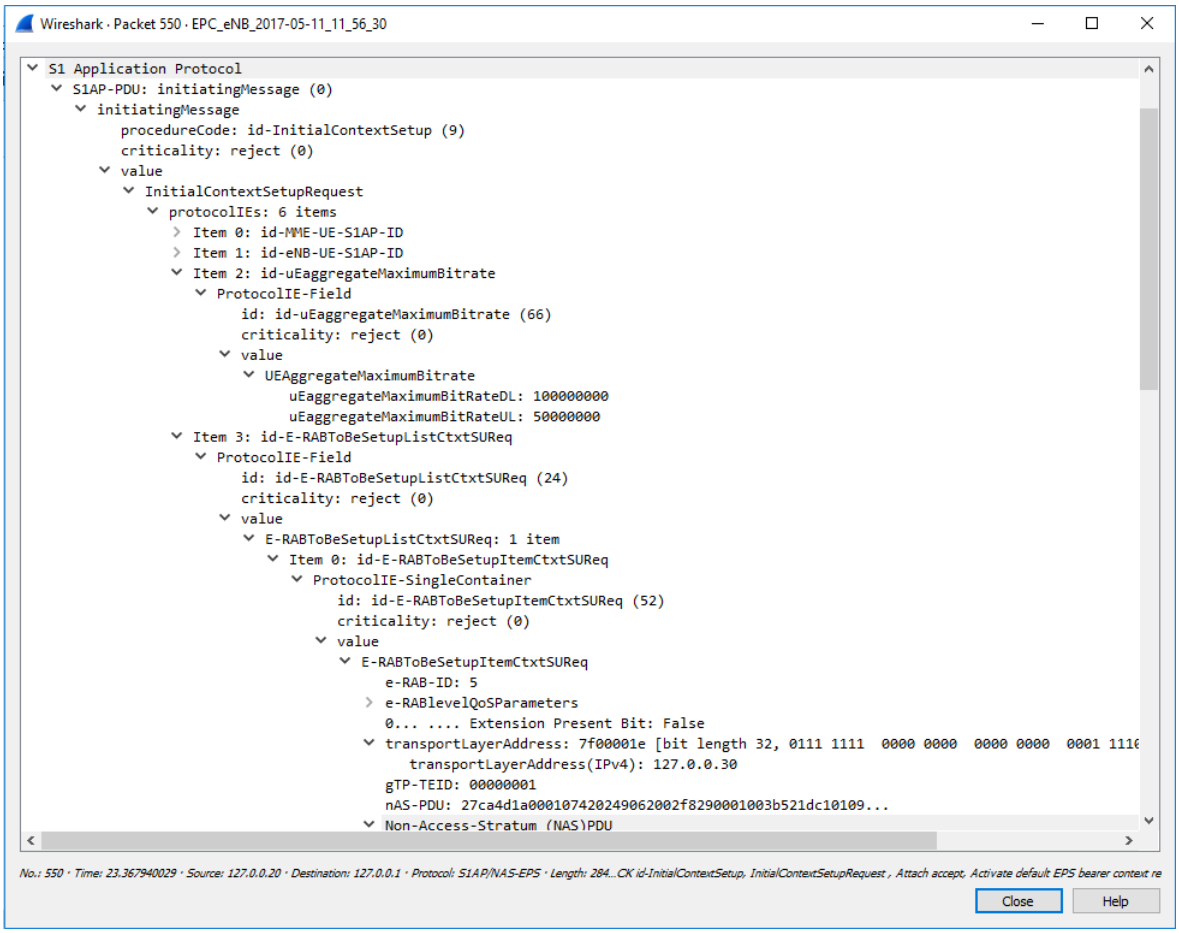



Figura 7.52: Mensaje *Initial Context Setup Request*.

Después de recibir el mensaje *Initial Context Setup Request*, el eNB realiza la configuración para la conexión S1 con el S-GW y el DRB con el UE. Previamente, el eNB realiza los procedimientos para establecer la seguridad en las comunicaciones, derivando las claves para gestionar la integridad y encriptación.

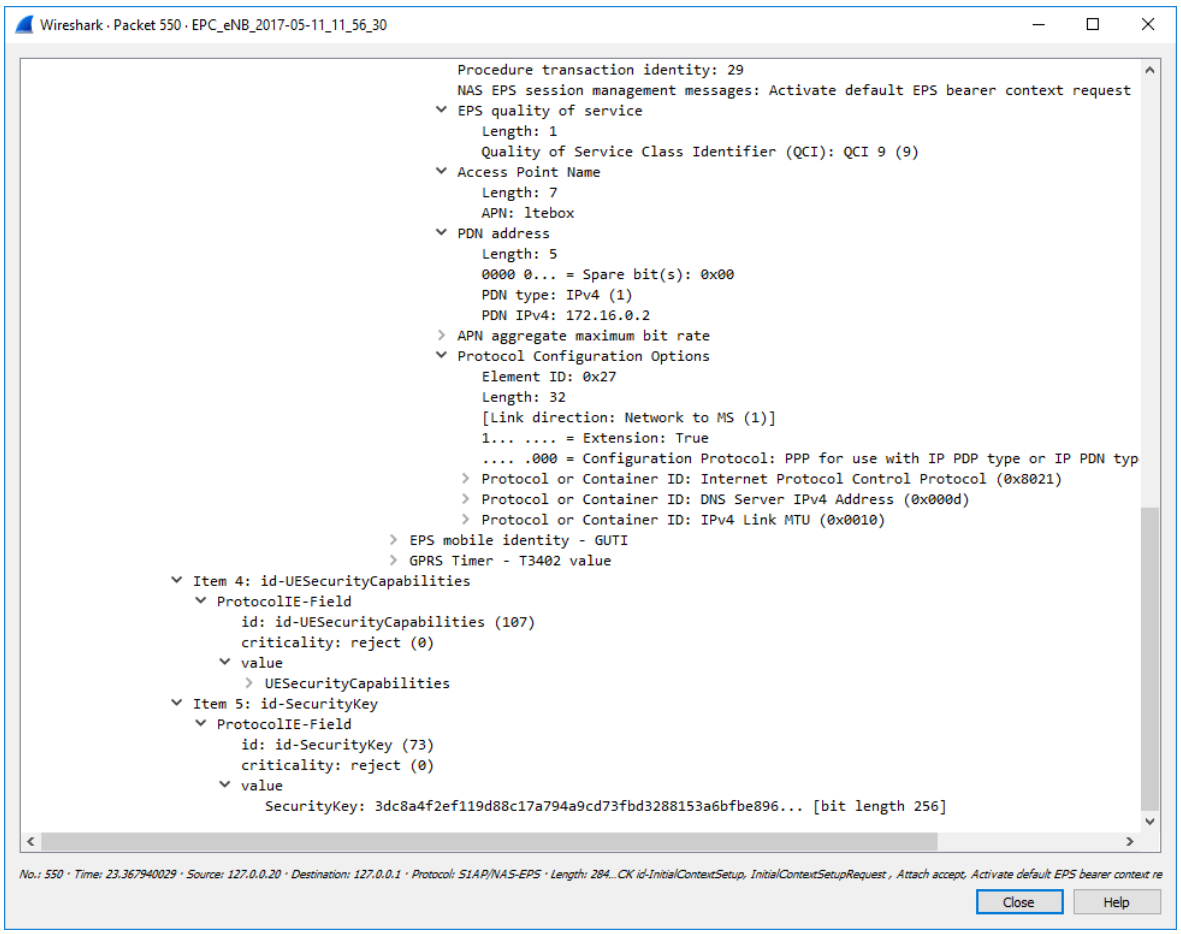



Figura 7.53: Mensaje *Initial Context Setup Request* (cont).

A continuación, cuando se ha generado el DRB, el eNB asigna un *S1 ENB TEID* de bajada para la conexión S1 y lo envía al MME, incluyéndolo en el mensaje *Initial Context Setup Response*. Este parámetro es enviado al S-GW, y éste informa al MME del establecimiento de la conexión S1 de bajada. Finalmente se crea un túnel de bajada S1 GTP-U desde el S-GW al eNB.

El mensaje *UE Capability Info* se envía desde el eNB hacia el MME, cuyo propósito es informar de las capacidades que tiene el UE y que se encuentran almacenadas en el eNB. Esta información proporcionada al MME, sobreescribe dicha información actualizando así las capacidades del UE.

### Desconexión UE - Red LTE

Para poder ver correctamente los mensajes para la desconexión del UE respecto de la red LTE. Estos mensajes pertenecen al móvil BQ que tiene insertada la USIM 11; forzaremos su salida al activar el módo avión.

El UE inicia el procedimiento de separación enviando un mensaje DETACH REQUEST. Podemos ver los detalles de este mensaje en la figura 7.54.

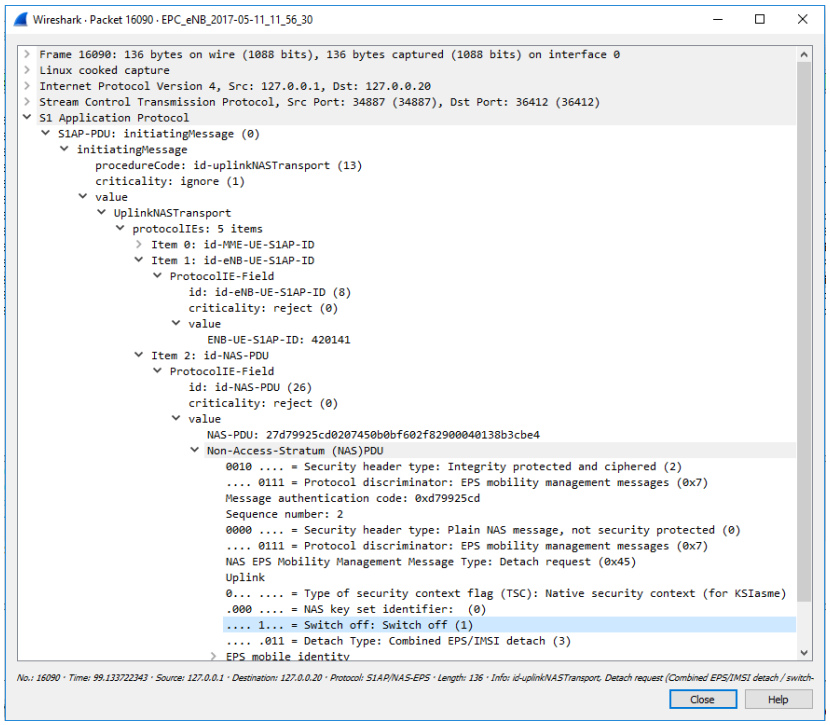



Figura 7.54: Mensaje *Detach Request*.

El MME envía al eNB el mensaje *UE Context Release Command* para liberar el contexto UE que está almacenado en dicho eNB. Una vez recibido este mensaje, el eNB envía un mensaje *RRC Connection Release* para eliminar la conexión RRC; finalmente se liberan los recursos asociados al UE y se elimina el contexto asignado al UE.

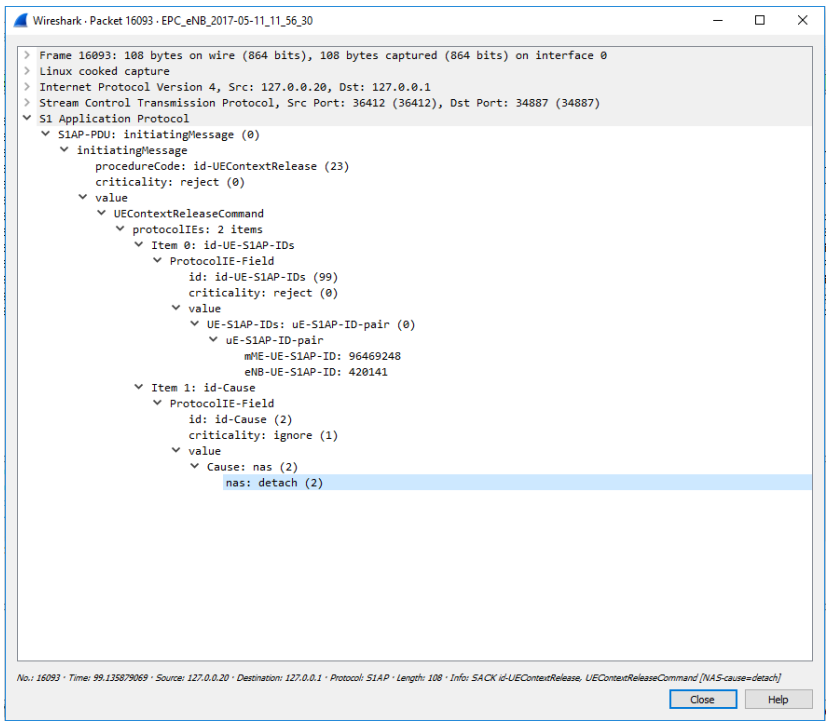



Figura 7.55: Mensaje *UE Context Release Command*.

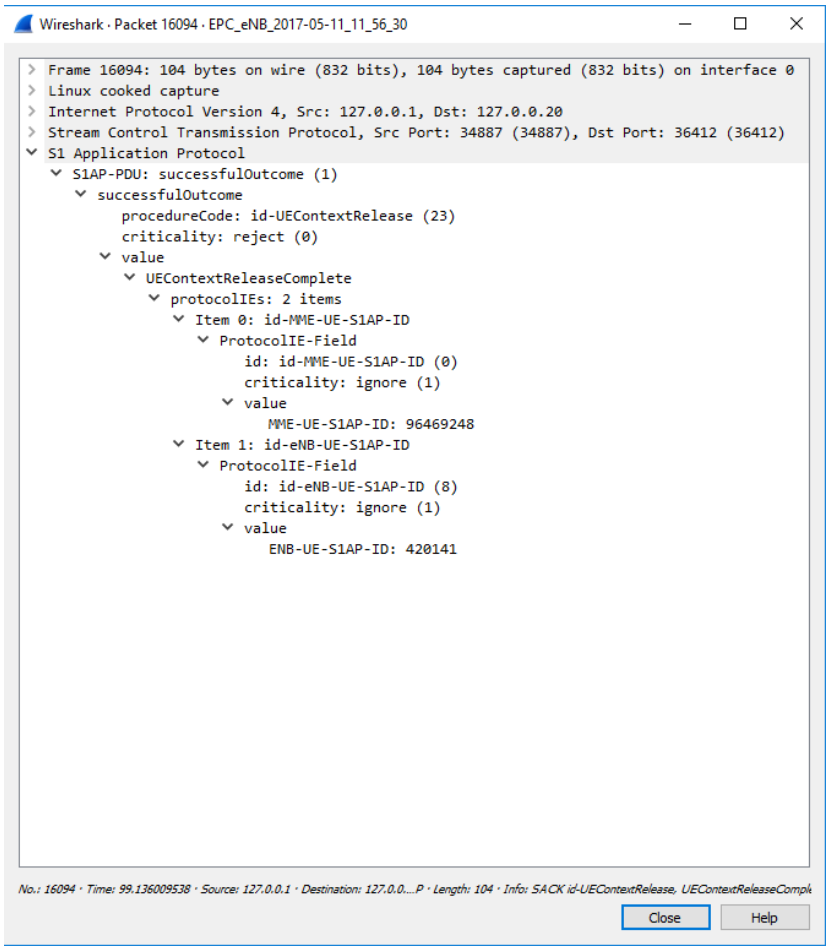



Figura 7.56: Mensaje *UE Context Release Complete*.

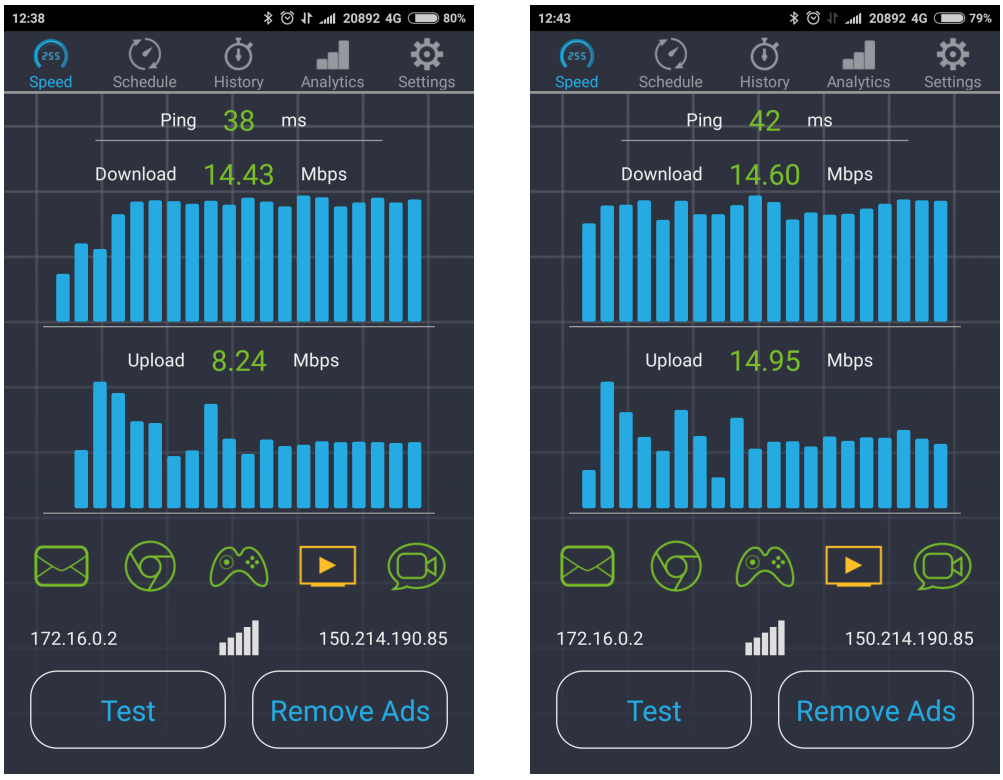
Una vez finalizada la liberación de recursos y la eliminación del contexto, el eNB envía un mensaje *UE Context Release Complete* como respuesta al mensaje anterior.

### 7.3.3. Pruebas de velocidad

En esta sección, se han realizado pruebas de velocidad con ambos teléfonos móviles. Para esta prueba hemos descargado la aplicación gratuita *"SpeedAnalytics"*.

En una primera prueba hemos realizado la conexión de cada móvil de manera independiente, y de forma que solamente esté un dispositivo conectado a nuestra red 4G en un instante dado.

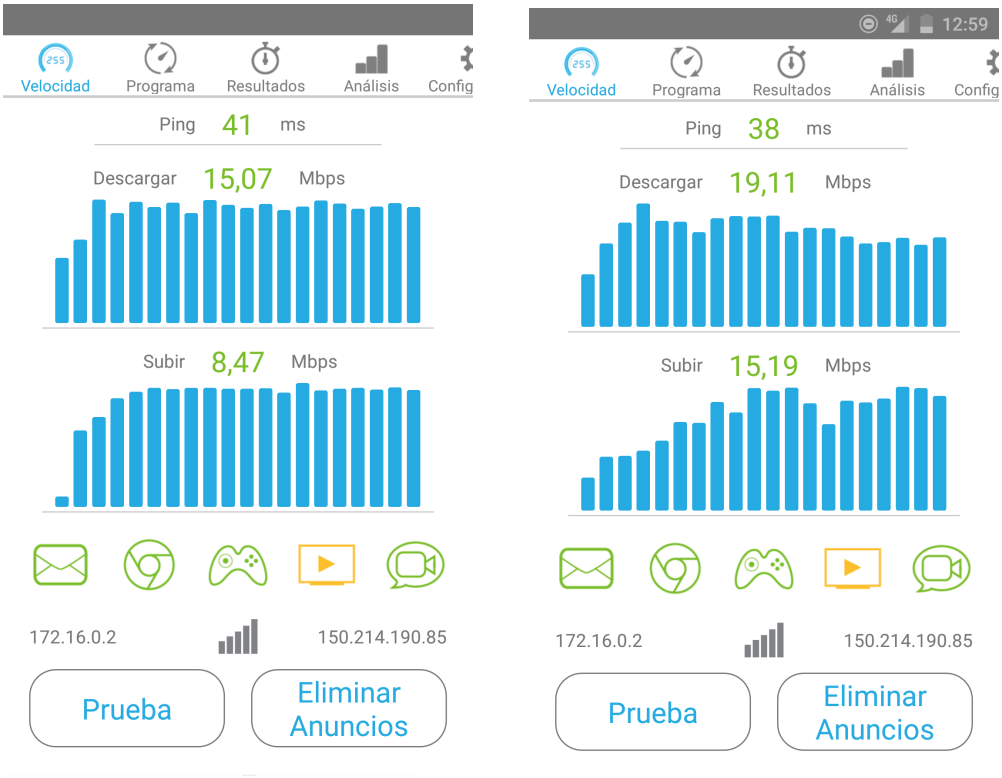
La figura 7.57 nos muestra dos resultados de esta prueba para el *smartphone* Xiaomi. En ella apreciamos que la dirección IP asignada por nuestra red al UE es 172.16.0.2. El equipo destino que responde a la prueba de ping tiene dirección IP 150.214.190.85, que corresponde con un equipo perteneciente a la Universidad de Granada (nombre de dominio "palas.ugr.es").




(a) Test: 1 (b) Test: 2

Figura 7.57: Test de velocidad UE: Xi.

En los test 7.57 obtenemos en ambos casos una velocidad media en el enlace descendente entorno a 14,5 Mbps; por otro lado, en el enlace ascendente observamos una variación considerable del primer test (8,24 Mbps) respecto del segundo; en éste último la velocidad media es de 14,95 Mbps con lo que tenemos un simetría en el ancho de banda para los dos enlaces. Otro parámetro que obtenemos con esta prueba es el retardo del ping, que eleva su valor respecto del primer test.




(a) Test: 1 (b) Test: 2

Figura 7.58: Test de velocidad UE: BQ.

Repitiendo el procedimiento anterior obtenemos los test para el dispositivo móvil BQ. En la figura 7.58, obtenemos dichos test de velocidad. En el primer test se ha obtenido una velocidad media más baja que en el segundo, tanto para el enlace ascendente como para el descendente. En el primero se ha obtenido una velocidad media en el enlace UL de 8,47 Mbps, mientras que en DL una velocidad de 15,07 Mbps, y un retardo de ping de 41 ms. Por otro lado, en el segundo test han aumentado las velocidades medias, en el enlace ascendente 15,19 Mbps, y en el enlace descendente 19,11 Mbps, y un retardo de 38 ms.

En cualquiera de los casos, las velocidades medias son similares para ambos enlaces. Además, observamos que el retardo también es parecido.

# Capítulo 8

# Conclusiones y líneas futuras

## 8.1. Introducción

En este capítulo se recogen las conclusiones finales, una vez que se ha dado por terminado el presente trabajo además de una serie de posibles desarrollos futuros.

## 8.2. Conclusiones

En el presente Trabajo Fin de Grado se ha implementado una estación base LTE utilizando el *software* libre Open Air Interface. Así, finalmente se han conseguido lograr los objetivos propuestos en un principio. Se ha conseguido realizar el estudio de distintos escenarios, así como la realización de distintas pruebas. Los escenarios desarrollados en el presente proyecto los resumiremos a continuación, destacando los aspectos más importantes.

En primer lugar, se ha querido implementar una red móvil celular LTE mediante dos máquinas virtuales, habiendo realizado la instalación y configuración en una de ellas de OAISIM y éste asumiendo el papel de eNB + UE; mientras en la otra se ha instalado y configurado el EPC y éste con las entidades de HSS + MME + S-GW. Con este procedimiento se han asentado los conocimientos sobre las redes móviles 4G, y los conocimientos obtenidos de instalación, configuración y funcionamiento del *software* OAI. Además, con esta parte del proyecto, se han comprobado los mensajes de señalización básicos cuando se activa un eNodeB y un UE.

En segundo lugar, fuimos un paso más allá queriendo trasladar la idea implementada en las máquinas virtuales a máquinas reales, utilizando un USRP para realizar la transmisión y recepción de datos, creando así nuestra propia red LTE y utilizando dispositivos móviles comerciales. En este escenario, hemos tenido que configurar la base de datos del HSS, introducir los datos correspondientes con las tarjetas USIM, realizar la programación correcta de cada una de éstas tarjetas, etcétera. Además, con respecto al primer escenario, se ha analizado la parte radio de nuestra red LTE; de otro modo, también se ha extendido y profundizado el análisis de trazas.

Cabe destacar que, una vez finalizado el proyecto, teniendo en cuenta que el paquete *software* OAI es una plataforma abierta y de código libre, está en desarrollo y hemos encontrado algunas debilidades.

- La primera de ellas, es que se trata de un paquete actualmente pensado para la investigación y la educación, por lo que no es lo suficientemente estable como para desplegar una red LTE de bajo coste.
- La segunda debilidad se produce en el escenario real. Si seleccionamos en el UE la opción de utilizar VoLTE (Voice over LTE), éste intenta realizar un solicitud para la parte de datos y la parte de conmutación de circuitos; en ese instante el MME finaliza su funcionamiento con un fallo, debido a que no soporta una conexión combinada (*combined attach*). Si desmarcamos esa opción del terminal móvil, nuestra red LTE funciona perfectamente.
- La tercera debilidad es su inestabilidad al establecer unos RB superiores a 25 para así conseguir distintos anchos de banda (15 y 20 MHz). En este caso, el sistema se vuelve muy inestable y normalmente el MME termina su funcionamiento cuando el UE intenta conectarse a la red. Aunque esta debilidad pensamos que se puede deber a la falta de prestaciones suficientes del PC y del USRP que estamos utilizando.

## 8.3. Líneas futuras

Como se ha comentado anteriormente, los objetivos propuestos desde un principio se han cumplido, aunque se pueden realizar numerosas vías de desarrollo futuro.

- Implementación de distintos escenarios en las máquinas virtuales, tales como realizar la instalación, configuración de las distintas entidades que componen una red móvil LTE en MV diferentes, ajustando así las prestaciones de cada una de ellas reduciendo los costes y evaluando su funcionamiento.
- Despliegue de más de una celda LTE y análisis de su comportamiento. Estudiar el mecanismo de movilidad o *handover* entre celdas. Será necesario la utilización de varios equipos SDR, realizar la correcta configuración de los equipos para que tengan un área de cobertura pequeña, llevar a cabo la configuración de las distintas entidades que componen una red LTE, etcétera.
- Implementación del escenario totalitario del uso de la plataforma OAI, en el que se utilizaría OAI UE, OAI eNB, OAI CN. Realizando la instalación en equipos distintos y de altas prestaciones, podría realizarse un análisis de la capa física y de la señalización intercambiada entre los nodos. Además, se realizaría una evaluación de las capacidades de OAI cuando intentamos maximizar los recursos prestados al UE, por ejemplo un ancho de banda de 20 MHz, y verificar si cumple con los requisitos de las especificaciones del 3GPP.

## 8.4. Valoración personal

Para finalizar, detallaremos una serie de apreciaciones y reflexiones personales acerca del desarrollo de este proyecto.

El Trabajo Fin de Grado representa la cumbre del éxito al finalizar la titulación. Es por ello que en él se deberá demostrar que hemos adquirido los conocimientos, competencias y aptitudes correctas que demuestran nuestra capacidad para adquirir, reunir diferentes datos, analizar información diversa dando así diferentes puntos de

vista, transmitir ideas, y establecer soluciones a los distintos problemas que pueden surgir. Finalmente, además de las aportaciones anteriores, nos brindan la capacidad de realizar estudios posteriores para incrementar nuestro potencial, formación y tener un alto grado de autonomía.

Este proyecto, intenta plasmar bastante bien un escenario comercial, orientado a la capacitación laboral en el ámbito de las redes móviles de cuarta generación. En plena revolución de la tecnología 4G, y a las puertas de la 5G, la formación sobre redes de última generación promete un futuro alentador. Por otra parte, la finalización de este proyecto ha supuesto la satisfacción tanto a nivel personal como profesional, debido al total desconocimiento del funcionamiento y complejidad de una red móvil LTE.

# Apéndice A

# Máquina virtual EPC

### Archivo */etc/network/interfaces* de la máquina virtual EPC

```
auto lo
iface lo inet loopback
# Configuración IP para los distintos interfaces de MV EPC
# Tarjeta para la conexión a Internet
auto eth0
iface eth0 inet static address 10.0.2.15
netmask 255.255.255.0
network 10.0.2.0
broadcast 10.0.2.255

# Tarjeta para la conexión con OAISIM
auto eth1
iface eth1 inet static address 192.168.1.1
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

## Archivos de configuración del HSS

### Archivo *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 man-
# dates that the Identity used is a valid FQDN for the peer. This para-
# meter can be omitted, in that case the framework will attempt to use
# system default value (as returned by hostname --fqdn).
Identity = "hss.5GLaboratory";

# 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 = "5GLaboratory";

# This parameter is mandatory, even if it is possible to disable TLS for
# peers connections. A valid certificate for this Diameter Identity is
# expected.
TLS_Cred = "/usr/local/etc/oai/freeDiameter/hss.cert.pem", "/usr/local
/etc/oai/freeDiameter/hss.key.pem";
TLS_CA = "/usr/local/etc/oai/freeDiameter/hss.cacert.pem";

# Disable use of TCP protocol (only listen and connect in SCTP)
# Default : TCP enabled
No_SCTP;

# This option is ignored if freeDiameter is compiled with DISABLE_SCTP
# option. Prefer TCP instead of SCTP for establishing new connections.
# This setting may be overwritten per peer in peer configuration blocs.
# Default : SCTP is attempted first.
Prefer_TCP;

# Disable use of IPv6 addresses (only IP)
# Default : IPv6 enabled
No_IPv6;

# Overwrite the number of SCTP streams. This value should be kept low,
# especially if you are using TLS over SCTP, because it consumes a lot
# of resources in that case. See tickets 19 and 27 for some additional
# details on this.
# Limit the number of SCTP streams
SCTP_streams = 3;

# By default, freeDiameter acts as a Diameter Relay Agent by forwarding
# all messages it cannot handle locally. This parameter disables this
# behavior.
NoRelay;

# Use RFC3588 method for TLS protection, where TLS is negociated after
```

```
# CER/CEA exchange is completed on the unsecure connection. The
# alternative is RFC6733 mechanism, where TLS protects also the CER/CEA
# exchange on a dedicated secure port.This parameter only affects
# outgoing connections. The setting can be also defined per-peer (see
# Peers configuration section).
# Default: use RFC6733 method with separate port for TLS.

#TLS_old_method;

# Number of parallel threads that will handle incoming application
# messages. This parameter may be deprecated later in favor of a dynamic
# number of threads depending on the load.
AppServThreads = 4;

# 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";

Port = 3868;
SecPort = 5868;

# -------- Extensions ---------

# Uncomment (and create rtd.conf) to specify routing table for this
# peer.
# LoadExtension = "rt_default.fdx" : "rtd.conf";

# Uncomment (and create acl.conf) to allow incoming connections from
# other peers.
LoadExtension = "acl_wl.fdx" : "/usr/local/etc/oai/freeDiameter
/acl.conf";

# Uncomment to display periodic state information
# LoadExtension = "dbg_monitor.fdx";

# Uncomment to enable an interactive Python interpreter session.
# (see doc/dbg_interactive.py.sample for more information)
# LoadExtension = "dbg_interactive.fdx";

# Load the RFC4005 dictionary objects
# LoadExtension = "dict_nasreq.fdx";

LoadExtension = "dict_nas_mipv6.fdx";
LoadExtension = "dict_s6a.fdx";

# Load RFC4072 dictionary objects
# LoadExtension = "dict_eap.fdx";
```

```
# Load the Diameter EAP server extension (requires diameap.conf)
# LoadExtension = "app_diameap.fdx" : "diameap.conf";

# Load the Accounting Server extension (requires app_acct.conf)
# LoadExtension = "app_acct.fdx" : "app_acct.conf";

# -------- Peers ---------

# The framework will actively attempt to establish and maintain a
# connection with the peers listed here. For only accepting incoming
# connections, see the acl_wl.fx extension.

ConnectPeer = "epc.5GLaboratory" { ConnectTo = "127.0.0.1"; No_TLS; };
```

### Archivo *hss.conf*

```
################################################################################
# Licensed to the OpenAirInterface (OAI) Software Alliance under one or
# more contributor license agreements.  See the NOTICE file distributed
# with this work for additional information regarding copyright
# ownership. The OpenAirInterface Software Alliance licenses this file to
# You under the Apache License, Version 2.0  (the "License"); you may
# not use this file except in compliance with the License. You may
# obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied. See the License for the specific language governing
# permissions and limitations under the License.
#-------------------------------------------------------------------------------
# For more information about the OpenAirInterface (OAI) Software
# Alliance:
#      contact@openairinterface.org
################################################################################

HSS :
{
    ///MySQL mandatory options
    MYSQL_server = "127.0.0.1";              # HSS S6a bind address
    MYSQL_user   = "root";                   # Database server login
    MYSQL_pass   = "5GLaboratory";           # Database server password
    MYSQL_db     = "oai_db";                 # Your database name

    ///HSS options
    # OP key matching your database
    OPERATOR_key   = "1006020f0a478bf6b699f15c062e42b3";
    # OP key matching your database
    #OPERATOR_key  = "11111111111111111111111111111111";
    # True random or only pseudo random (for subscriber vector
    # generation)

    RANDOM = "true";

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

## Archivos de configuración del MME

### Archivo *mme_fd.conf*

```
# -------- Local ---------

# Uncomment if the framework cannot resolv it.
    Identity  = "epc.5GLaboratory";
    Realm     = "5GLaboratory";

# TLS configuration (see previous section)
    TLS_Cred = "/usr/local/etc/oai/freeDiameter/mme.cert.pem",
               "/usr/local/etc/oai/freeDiameter/mme.key.pem";
    TLS_CA   = "/usr/local/etc/oai/freeDiameter/mme.cacert.pem";

# Disable use of TCP protocol (only listen and connect in SCTP)
# Default : TCP enabled
    No_SCTP;

# This option is ignored if freeDiameter is compiled with DISABLE_SCTP
# option. Prefer TCP instead of SCTP for establishing new connections.
# This setting may be overwritten per peer in peer configuration blocs.
# Default : SCTP is attempted first.
    Prefer_TCP;
    No_IPv6;

# Overwrite the number of SCTP streams. This value should be kept low,
# especially if you are using TLS over SCTP, because it consumes a lot
# of resources in that case. See tickets 19 and 27 for some additional
# details on this.
# Limit the number of SCTP streams
    SCTP_streams = 3;


# By default, freeDiameter acts as a Diameter Relay Agent by forwarding
# all messages it cannot handle locally. This parameter disables this
# behavior.
    NoRelay;

# Use RFC3588 method for TLS protection, where TLS is negociated after
# CER/CEA exchange is completed on the unsecure connection. The
# alternative is RFC6733 mechanism, where
# TLS protects also the CER/CEA exchange on a dedicated secure port.
# This parameter only affects outgoing connections.
# The setting can be also defined per-peer (see Peers configuration
# section).
# Default: use RFC6733 method with separate port for TLS.
#TLS_old_method;
    AppServThreads = 4;

# 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 = ;

Port    = 3870;
SecPort = 5870;

# -------- Extensions ---------

# Uncomment (and create rtd.conf) to specify routing table for this
# peer.
#LoadExtension = "rt_default.fdx" : "rtd.conf";

# Uncomment (and create acl.conf) to allow incoming connections from
# other peers.
#LoadExtension = "acl_wl.fdx" : "acl.conf";

# Uncomment to display periodic state information
#LoadExtension = "dbg_monitor.fdx";

# Uncomment to enable an interactive Python interpreter session.
# (see doc/dbg_interactive.py.sample for more information)
#LoadExtension = "dbg_interactive.fdx";

# Load the RFC4005 dictionary objects
#LoadExtension = "dict_nasreq.fdx";

    LoadExtension = "dict_nas_mipv6.fdx";
    LoadExtension = "dict_s6a.fdx";

# Load RFC4072 dictionary objects
#LoadExtension = "dict_eap.fdx";

# Load the Diameter EAP server extension (requires diameap.conf)
#LoadExtension = "app_diameap.fdx" : "diameap.conf";

# Load the Accounting Server extension (requires app_acct.conf)
#LoadExtension = "app_acct.fdx" : "app_acct.conf";

# -------- Peers ---------
# The framework will actively attempt to establish and maintain a
# connection with the peers listed here. For only accepting incoming
# connections, see the acl_wl.fx extension.
# ConnectPeer
# Declare a remote peer to which this peer must maintain a connection.
# In addition, this allows specifying non-default parameters for this
# peer only (for example disable SCTP with this peer, or use RFC3588-
# flavour TLS). Note that by default, if a peer is not listed as a
```

```
# ConnectPeer entry, an incoming connection from this peer will be
# rejected. If you want to accept incoming connections from other peers,
# see the acl_wl.fdx? extension which allows exactly this.

    ConnectPeer= "hss.5GLaboratory" { ConnectTo = "127.0.0.1"; No_SCTP ;
    No_IPv6; Prefer_TCP; No_TLS; port = 3868;  realm = "5GLaboratory";};
```

### Archivo *mme.conf*

```
################################################################################
#
# Licensed to the OpenAirInterface (OAI) Software Alliance under one or
# more contributor license agreements.  See the NOTICE file distributed
# with this work for additional information regarding copyright
# ownership. The OpenAirInterface Software Alliance licenses this file
# to You under the Apache License, Version 2.0  (the "License"); you may
# not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied. See the License for the specific language governing
# permissions and limitations under the License.
#-------------------------------------------------------------------------------
# For more information about the OpenAirInterface (OAI) Software
# Alliance:
#      contact@openairinterface.org
################################################################################


MME :
{
    REALM                   = "5GLaboratory";        # YOUR REALM HERE

    # Define the limits of the system in terms of served eNB and served
    # UE. When the limits will be reached, overload procedure will take
    # place.
    MAXENB                  = 2;                     # power of 2
    MAXUE                   = 16;                    # power of 2
    RELATIVE_CAPACITY       = 10;

    EMERGENCY_ATTACH_SUPPORTED          = "no";
    UNAUTHENTICATED_IMSI_SUPPORTED      = "no";

    # EPS network feature support
    # DO NOT CHANGE
    EPS_NETWORK_FEATURE_SUPPORT_IMS_VOICE_OVER_PS_SESSION_IN_S1  = "no";
    EPS_NETWORK_FEATURE_SUPPORT_EMERGENCY_BEARER_SERVICES_IN_S1_MODE =
    "no";
    EPS_NETWORK_FEATURE_SUPPORT_LOCATION_SERVICES_VIA_EPC        = "no";
    EPS_NETWORK_FEATURE_SUPPORT_EXTENDED_SERVICE_REQUEST         = "no";

    # Display statistics about whole system (expressed in seconds)
    MME_STATISTIC_TIMER = 10;
```

```
IP_CAPABILITY  = "IPV4V6";   # UNUSED, TODO

INTERTASK_INTERFACE :
{
    # max queue size per task
    ITTI_QUEUE_SIZE  = 2000000;
};

S6A :
{ # YOUR MME freeDiameter config file path
    S6A_CONF  = "/usr/local/etc/oai/freeDiameter/mme_fd.conf";
    HSS_HOSTNAME  = "hss";                    # THE HSS HOSTNAME
};

# ------- SCTP definitions
SCTP :
{
    # Number of streams to use in input/output
    SCTP_INSTREAMS  = 8;
    SCTP_OUTSTREAMS = 8;
};

# ------- S1AP definitions
S1AP :
{
    # outcome drop timer value (seconds)
    S1AP_OUTCOME_TIMER = 10;
};

# ------- MME served GUMMEIs
# MME code DEFAULT  size = 8 bits
# MME GROUP ID size      = 16 bits
# YOUR GUMMEI CONFIG HERE
GUMMEI_LIST = (
     {MCC="208" ; MNC="93"; 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
# YOUR TAI CONFIG HERE
TAI_LIST = (
     {MCC="208" ; MNC="93";  TAC = "1"; }
);


NAS :
```

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

    # EMM TIMERS
    # T3402 start:
    # At attach failure and the attempt counter is equal to 5.
    # At tracking area updating failure and the attempt counter is
    # equal to 5.
    # T3402 stop:
    # ATTACH REQUEST sent, TRACKING AREA REQUEST sent.
    # On expiry:
    # Initiation of the attach procedure, if still required or TAU
    # procedure attached for emergency bearer services.
    T3402 =  12;         # in minutes (default is 12 minutes)

    # T3412 start:
    # In EMM-REGISTERED, when EMM-CONNECTED mode is left.
    # T3412 stop:
    # When entering state EMM-DEREGISTERED or when entering EMM-
    # CONNECTED mode.
    # On expiry:
    # Initiation of the periodic TAU procedure if the UE is not
    # attached for emergency bearer services. Implicit detach from
    # network if the UE is attached for emergency bearer services.
    # in minutes (default is 54 minutes, network dependent)
    T3412 =  54;

    # T3422 start: DETACH REQUEST sent
    # T3422 stop: DETACH ACCEPT received
    # ON THE 1st, 2nd, 3rd, 4th EXPIRY: Retransmission of DETACH
    # REQUEST in seconds (default is 6s)
    T3422   =  6;

    # T3450 start:
    # ATTACH ACCEPT sent, TRACKING AREA UPDATE ACCEPT sent with
    # GUTI, TRACKING
    # AREA UPDATE ACCEPT sent with TMSI,
    # GUTI REALLOCATION COMMAND sent
    # T3450 stop:
    # ATTACH COMPLETE received, TRACKING AREA UPDATE COMPLETE
    # received, GUTI
    # REALLOCATION COMPLETE received
    # ON THE 1st, 2nd, 3rd, 4th EXPIRY: Retransmission of the same
    # message type in seconds (default is 6s)
    T3450   =  6;
```

```
    # T3460 start: AUTHENTICATION REQUEST sent, SECURITY MODE
    # COMMAND sent
    # T3460 stop:
    # AUTHENTICATION RESPONSE received, AUTHENTICATION FAILURE
    # received, SECURITY MODE COMPLETE received, SECURITY MODE
    # REJECT received ON THE 1st, 2nd, 3rd, 4th EXPIRY:
    # Retransmission of the same message type in seconds (default is
    # 6s)
    T3460 =  6;

    # T3470 start: IDENTITY REQUEST sent
    # T3470 stop: IDENTITY RESPONSE received
    # ON THE 1st, 2nd, 3rd, 4th EXPIRY: Retransmission of IDENTITY
    # REQUEST in seconds (default is 6s)
    T3470 =  6;

    # ESM TIMERS
    # T3485 start: ACTIVATE DEFAULT EPS BEARER CONTEXT REQUEST sent,
    # ACTIVATE DEDICATED EPS BEARER CONTEXT REQUEST sent
    # T3485 stop:  ACTIVATE DEFAULT EPS BEARER CONTEXT ACCEPT
    # received or  ACTIVATE DEFAULT EPS BEARER CONTEXT REJECT
    # received or  ACTIVATE DEDICATED EPS BEARER CONTEXT ACCEPT
    # received or  ACTIVATE DEDICATED EPS BEARER CONTEXT REJECT
    # received ON THE 1st, 2nd, 3rd, 4th EXPIRY: Retransmission of
    # the same message in seconds (default is 8s)
    T3485   =  8;
    # UNUSED in seconds (default is 8s)
    T3486   =  8;

    # T3489 start: ESM INFORMATION REQUEST sent
    # T3489 stop:  ESM INFORMATION RESPONSE received
    # Retransmission of ESM INFORMATION REQUEST on 1st and 2nd
    # expiry only in seconds (default is 4s)
    T3489 =  4;
    # UNUSED in seconds (default is 8s)
    T3495   =  8;

    # NON STANDARD FEATURES
    # Reject Tracking Area Update
    # Leave it at "yes" (TAU TODO)
    FORCE_REJECT_TAU = "yes";
    # Reject Service Request
    # Leave it at "yes" (SR TODO)
    FORCE_REJECT_SR     = "yes";
    DISABLE_ESM_INFORMATION_PROCEDURE     = "yes";
};

NETWORK_INTERFACES :
{
```

```
        # MME binded interface for S1-C or S1-MME  communication (S1AP),
        # can be ethernet interface, virtual ethernet interface, we
        # don't advise wireless interfaces
        # YOUR NETWORK CONFIG HERE
        MME_INTERFACE_NAME_FOR_S1_MME         = "eth1";
        MME_IPV4_ADDRESS_FOR_S1_MME           = "192.168.1.1/24";

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

    LOGGING :
    {
        # OUTPUT choice in { "CONSOLE", "SYSLOG", `path to file`",
        # "`IPv4@`:`TCP port num`"}
        # `path to file` must start with '.' or '/'
        # if TCP stream choice, then you can easily dump the traffic on
        # the remote or local host: nc -l `TCP port num` > received.txt
        OUTPUT                      = "CONSOLE";
        #OUTPUT                     = "SYSLOG";
        #OUTPUT                     = "/tmp/mme.log";
        #OUTPUT                     = "127.0.0.1:5656";

        # THREAD_SAFE choice in { "yes", "no" } means use of thread safe
        # intermediate buffer then a single thread pick each message log
        # one by one to flush it to the chosen output
        THREAD_SAFE                 = "no";

        # COLOR choice in { "yes", "no" } means use of ANSI styling
        # codes or no
        COLOR                       = "yes";

        # Log level choice in { "EMERGENCY", "ALERT", "CRITICAL",
        # "ERROR",
        # "WARNING", "NOTICE", "INFO", "DEBUG", "TRACE"}
        SCTP_LOG_LEVEL                = "NOTICE";
        S11_LOG_LEVEL                 = "TRACE";
        GTPV2C_LOG_LEVEL              = "TRACE";
        UDP_LOG_LEVEL                 = "TRACE";
        S1AP_LOG_LEVEL                = "TRACE";
        NAS_LOG_LEVEL                 = "TRACE";
        MME_APP_LOG_LEVEL             = "TRACE";
        S6A_LOG_LEVEL                 = "TRACE";
        SECU_LOG_LEVEL                = "TRACE";
        UTIL_LOG_LEVEL                = "TRACE";
        MSC_LOG_LEVEL                 = "WARNING";
        ITTI_LOG_LEVEL                = "WARNING";
```

```
        XML_LOG_LEVEL                  = "TRACE";
        MME_SCENARIO_PLAYER_LOG_LEVEL = "TRACE";

        # ASN1 VERBOSITY: none, info, annoying
        # for S1AP protocol
        ASN1_VERBOSITY    = "none";
    };

    S-GW_LIST_SELECTION = (
         {
         ID="tac-lb01.tac-hb00.tac.epc.mnc092.mcc208.5GLaboratory";
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb01.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb02.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb03.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb04.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb05.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb06.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb07.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb08.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb09.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb0a.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb0b.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb0c.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb0d.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb0e.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";},
         {ID="tac-lb0f.tac-hb00.tac.epc.mnc093.mcc208.3gppnetwork.org" ;
         SGW_IPV4_ADDRESS_FOR_S11="127.0.11.2/8";}
    );

};
```

### Archivo *acl.conf*

```
# Configuration file for the peer whitelist extension.
# This extension is meant to allow connection from remote peers, without
# actively maintaining this connection ourselves (as it would be the
# case by declaring the peer in a ConnectPeer directive). The format of
# this file is very simple. It contains a list of peer names separated
# by spaces or newlines.
#
# The peer name must be a fqdn. We allow also a special "*" character as
# the first label of the fqdn, to allow all fqdn with the same domain
# name.
# Example: *.example.net will allow host1.example.net and
# host2.example.net
#
# At the beginning of a line, the following flags are allowed (case
# sensitive)
# -- either or both can appear:
# 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   *.5GLaboratory
```

## Archivos de configuración del S-GW

### Archivo *spgw.conf*

```
################################################################################
#
# Licensed to the OpenAirInterface (OAI) Software Alliance under one or
# more contributor license agreements.  See the NOTICE file distributed
# with this work for additional information regarding copyright
# ownership. The OpenAirInterface Software Alliance licenses this file
# to You under the Apache License, Version 2.0  (the "License"); you may
# not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied. See the License for the specific language governing
# permissions and limitations under the License.
#-------------------------------------------------------------------------------
# For more information about the OpenAirInterface (OAI) Software
# Alliance:
#      contact@openairinterface.org
################################################################################

S-GW :
{
    NETWORK_INTERFACES :
    {
        # S-GW binded interface for S11 communication (GTPV2-C), if none
        # selected the ITTI message interface is used
        # STRING, interface name, YOUR NETWORK CONFIG HERE
            SGW_INTERFACE_NAME_FOR_S11              = "lo";
            SGW_IPV4_ADDRESS_FOR_S11                = "127.0.11.2/8";

        # S-GW binded interface for S1-U communication (GTPV1-U) can be
        # ethernet interface, virtual ethernet interface, we don't
        # advise wireless interfaces
            SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP    = "eth1";
            SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP      = "192.168.1.1/24";
            SGW_IPV4_PORT_FOR_S1U_S12_S4_UP         = 2152;

        # INTEGER, port number, PREFER NOT CHANGE UNLESS YOU KNOW WHAT
        # YOU ARE DOING

        # S-GW binded interface for S5 or S8 communication, not
        # implemented, so leave it to none
        # STRING, interface name, DO NOT CHANGE (NOT IMPLEMENTED YET)
        # STRING, CIDR, DO NOT CHANGE (NOT IMPLEMENTED YET)
```

```
            SGW_INTERFACE_NAME_FOR_S5_S8_UP         = "none";
            SGW_IPV4_ADDRESS_FOR_S5_S8_UP           = "0.0.0.0/24";

    };

    INTERTASK_INTERFACE :
    {
        # max queue size per task
        ITTI_QUEUE_SIZE = 2000000;                          # INTEGER
    };

    LOGGING :
    {
        # OUTPUT choice in { "CONSOLE", "SYSLOG", `path to file`",
        # "`IPv4@`:`TCP port num`"}
        # `path to file` must start with '.' or '/'
        # if TCP stream choice, then you can easily dump the traffic on
        # the remote or local host: nc -l `TCP port num` > received.txt
            OUTPUT             = "CONSOLE";       # see 3 lines above
           #OUTPUT             = "SYSLOG";        # see 4 lines above
           #OUTPUT             = "/tmp/spgw.log"; # see 5 lines above
           #OUTPUT             = "127.0.0.1:5656";# see 6 lines above

        # THREAD_SAFE choice in { "yes", "no" } means use of thread safe
        # intermediate buffer then a single thread pick each message log
        # one by one to flush it to the chosen output
            THREAD_SAFE        = "no";

        # COLOR choice in { "yes", "no" } means use of ANSI styling
        # codes or no
            COLOR              = "yes";

        # Log level choice in { "EMERGENCY", "ALERT", "CRITICAL",
        # "ERROR", "WARNING", "NOTICE", "INFO", "DEBUG", "TRACE"}
            ASYNC_SYSTEM       = "TRACE";
            UDP_LOG_LEVEL      = "TRACE";
            GTPV1U_LOG_LEVEL   = "TRACE";
            GTPV2C_LOG_LEVEL   = "TRACE";
            SPGW_APP_LOG_LEVEL = "TRACE";
            S11_LOG_LEVEL      = "TRACE";
            UTIL_LOG_LEVEL     = "TRACE";
            ITTI_LOG_LEVEL     = "WARNING";
    };
};

P-GW =
{
    NETWORK_INTERFACES :
    {
        # P-GW binded interface for S5 or S8 communication, not
```

```
    # implemented, so leave it to none
    # STRING, interface name, DO NOT CHANGE (NOT IMPLEMENTED YET)
        PGW_INTERFACE_NAME_FOR_S5_S8          = "none";

    # P-GW binded interface for SGI (egress/ingress internet
    # traffic)
        PGW_INTERFACE_NAME_FOR_SGI            = "eth0";
    # STRING, YOUR NETWORK CONFIG HERE
        PGW_MASQUERADE_SGI                    = "yes";
    # STRING, {"yes", "no"}. YOUR NETWORK CONFIG HERE, will do NAT
    # for you  if you put "yes".
# PGW_IPV4_ADDRESS_FOR_SGI       = "192.168.248.2/24";
        UE_TCP_MSS_CLAMPING           = "no";   # STRING,
        {"yes", "no"}.
};

# Pool of UE assigned IP addresses
# Do not make IP pools overlap
# first IPv4 address X.Y.Z.1 is reserved for GTP network device on
# SPGW. Normally no more than 16 pools allowed, but since recent GTP
# kernel module use, only one pool allowed (TODO).
    IP_ADDRESS_POOL :
    {
    IPV4_LIST = (
                  "172.16.0.0/12"   # STRING, CIDR, YOUR NETWORK
                   # CONFIG HERE.
                );
    };

# DNS address communicated to UEs
DEFAULT_DNS_IPV4_ADDRESS     = "8.8.8.8"; # YOUR NETWORK CONFIG HERE
DEFAULT_DNS_SEC_IPV4_ADDRESS = "8.8.4.4"; # YOUR NETWORK CONFIG HERE

# Non standard feature, normally should be set to "no", but you may
# need to set to yes for UE that do not explicitly request a PDN
# address through NAS signalling
# STRING,{"yes", "no"}.
    FORCE_PUSH_PROTOCOL_CONFIGURATION_OPTIONS = "no";
    UE_MTU                                    = 1500     # INTEGER
# STRING {"NO_GTP_KERNEL_AVAILABLE", "GTP_KERNEL_MODULE",
# "GTP_KERNEL"}.
# In a 8container you may not be able to unload/load kernel modules.
    GTPV1U_REALIZATION                        = "GTP_KERNEL_MODULE";

PCEF :
{
# STRING, {"yes", "no"}, if yes then all parameters bellow
# will/should be taken into account
      PCEF_ENABLED                            = "yes";
# STRING, {"yes", "no"}, TODO, should finally work for egress but
```

```
    # only on ingress bearers and not on ingress SDF flows
          TRAFFIC_SHAPPING_ENABLED                = "yes";
    # STRING, {"yes", "no"}, TCP explicit congestion notification
          TCP_ECN_ENABLED                         = "yes";
    # INTEGER [ 0..n], SDF identifier (Please check with enum sdf_id_t
    # in
    # pgw_pcef_emulation.h,
          AUTOMATIC_PUSH_DEDICATED_BEARER_PCC_RULE= 0;
    # SDF identifier for default bearer
        DEFAULT_BEARER_STATIC_PCC_RULE = 31;
    # List of SDF identifiers
        PUSH_STATIC_PCC_RULES          = (31);

    # Waiting for HSS APN-AMBR IE ...
    # Maximum UL bandwidth that can be used by non guaranteed bit rate
    # traffic in Kbits/seconds.
           APN_AMBR_UL = 500000;
    # Maximum DL bandwidth that can be used by non guaranteed bit rate
    # traffic in Kbits/seconds.
           APN_AMBR_DL = 500000;
    };
};
```

# Apéndice B

# Máquina virtual OAISIM

### Archivo */etc/network/interfaces* de la máquina virtual OAISIM

```
# Configuración IP para los distintos interfaces de MV OAISIM
# Tarjeta para la conexión con el EPC
auto eth1
iface eth1 inet static address 192.168.1.2
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

## Archivo de configuración de OAISIM

### Archivo *enb.band7.generic.oaisim.local_mme.conf*

```
Active_eNBs             = ( "eNB_5GLaboratory_OAISIM");
# Asn1_verbosity, choice in: none, info, annoying
Asn1_verbosity          = "none";

eNBs =
(
    {
    ////////// Identification parameters:
    eNB_ID              =  0xe00;
    cell_type           =  "CELL_MACRO_ENB";
    eNB_name            =  "eNB_5GLaboratory_OAISIM";

    // Tracking area code, 0x0000 and 0xfffe are reserved values
    tracking_area_code   =  "1";
    mobile_country_code  =  "208";
    mobile_network_code  =  "93";

    ////////// Physical parameters:
    component_carriers = (
     {
        frame_type                                  = "FDD";
        tdd_config                                  = 3;
        tdd_config_s                                = 0;
        prefix_type                                 = "NORMAL";
        eutra_band                                  = 7;
        downlink_frequency                          = 2680000000L;
        uplink_frequency_offset                     = -120000000;
        Nid_cell                                    = 0;
        N_RB_DL                                     = 25;
        Nid_cell_mbsfn                              = 0;
        nb_antenna_ports                            = 2;
        nb_antennas_tx                              = 2;
        nb_antennas_rx                              = 2;
        tx_gain                                     = 25;
        rx_gain                                     = 20;

        prach_root                                  = 0;
        prach_config_index                          = 0;
        prach_high_speed                            = "DISABLE";
        prach_zero_correlation                      = 1;
        prach_freq_offset                           = 2;

        pucch_delta_shift                           = 1;
        pucch_nRB_CQI                               = 1;
        pucch_nCS_AN                                = 0;
        pucch_n1_AN                                 = 32;
        pdsch_referenceSignalPower                  = 0;
```

```
pdsch_p_b                                       = 0;
pusch_n_SB                                      = 1;
pusch_enable64QAM                               = "DISABLE";
pusch_hoppingMode                               = "interSubFrame";
pusch_hoppingOffset                             = 0;
pusch_groupHoppingEnabled                       = "ENABLE";
pusch_groupAssignment                           = 0;
pusch_sequenceHoppingEnabled                    = "DISABLE";
pusch_nDMRS1                                    = 0;
phich_duration                                  = "NORMAL";
phich_resource                                  = "ONESIXTH";
srs_enable                                      = "DISABLE";
/*  srs_BandwidthConfig                         =;
srs_SubframeConfig                              =;
srs_ackNackST                                   =;
srs_MaxUpPts                                    =;*/
pusch_p0_Nominal                                = -108;
pusch_alpha                                     = "AL1";
pucch_p0_Nominal                                = -108;
msg3_delta_Preamble                             = 6;

pucch_deltaF_Format1                            = "deltaF2";
pucch_deltaF_Format1b                           = "deltaF3";
pucch_deltaF_Format2                            = "deltaF0";
pucch_deltaF_Format2a                           = "deltaF0";
pucch_deltaF_Format2b                           = "deltaF0";

rach_numberOfRA_Preambles                       = 64;
rach_preamblesGroupAConfig                      = "DISABLE";

/*rach_sizeOfRA_PreamblesGroupA                 = ;
rach_messageSizeGroupA                          = ;
rach_messagePowerOffsetGroupB                   = ;
*/
rach_powerRampingStep                           = 2;
rach_preambleInitialReceivedTargetPower         = -100;
rach_preambleTransMax                           = 10;
rach_raResponseWindowSize                       = 10;
rach_macContentionResolutionTimer               = 48;
rach_maxHARQ_Msg3Tx                             = 4;
pcch_default_PagingCycle                        = 128;
pcch_nB                                         = "oneT";
bcch_modificationPeriodCoeff                    = 2;
ue_TimersAndConstants_t300                      = 1000;
ue_TimersAndConstants_t301                      = 1000;
ue_TimersAndConstants_t310                      = 1000;
ue_TimersAndConstants_t311                      = 10000;
ue_TimersAndConstants_n310                      = 20;
ue_TimersAndConstants_n311                      = 1;
ue_TransmissionMode                             = 2;
```

```
    }
 );

srb1_parameters :
{
    # timer_poll_retransmit = (ms) [5, 10, 15, 20,... 250, 300, 350,
    # ... 500]
    timer_poll_retransmit    = 80;

    # timer_reordering = (ms) [0,5, ... 100, 110, 120, ... ,200]
    timer_reordering         = 35;

    # timer_reordering = (ms) [0,5, ... 250, 300, 350, ... ,500]
    timer_status_prohibit    = 0;

    # poll_pdu = [4, 8, 16, 32 , 64, 128, 256, infinity(>10000)]
    poll_pdu                 =  4;

    # poll_byte = (kB)
    # [25,50,75,100,125,250,375,500,750,1000,1250,1500,
    #                   2000,3000,infinity(>10000)]
    poll_byte                =  99999;

    # max_retx_threshold     = [1, 2, 3, 4 , 6, 8, 16, 32]
    max_retx_threshold       =  4;
}

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

////////// MME parameters:
mme_ip_address      = ( { ipv4       = "192.168.31.1";
                          ipv6       = "192:168:30::17";
                          active     = "yes";
                          preference = "ipv4";
                        }
                      );

NETWORK_INTERFACES :
{
    ENB_INTERFACE_NAME_FOR_S1_MME            = "eth1";
    ENB_IPV4_ADDRESS_FOR_S1_MME              = "192.168.1.2/24";

    ENB_INTERFACE_NAME_FOR_S1U               = "eth1";
    ENB_IPV4_ADDRESS_FOR_S1U                 = "192.168.1.2/24";
```

```
        ENB_PORT_FOR_S1U                        = 2152; # Spec 2152
    };

    log_config :
    {
    global_log_level                    ="trace";
    global_log_verbosity                ="medium";
    hw_log_level                        ="info";
    hw_log_verbosity                    ="medium";
    phy_log_level                       ="trace";
    phy_log_verbosity                   ="medium";
    mac_log_level                       ="trace";
    mac_log_verbosity                   ="medium";
    rlc_log_level                       ="trace";
    rlc_log_verbosity                   ="medium";
    pdcp_log_level                      ="trace";
    pdcp_log_verbosity                  ="medium";
    rrc_log_level                       ="trace";
    rrc_log_verbosity                   ="medium";
    gtpu_log_level                      ="debug";
    gtpu_log_verbosity                  ="medium";
    udp_log_level                       ="debug";
    udp_log_verbosity                   ="medium";
    osa_log_level                       ="debug";
    osa_log_verbosity                   ="low";
   };
  }
);
```

# Apéndice C

# Manual del usuario OAI: CN y eNB

Este anexo, muestra los pasos a seguir para la puesta en funcionamiento mediante la plataforma OAI, del CN, y del eNB. Suponemos que los equipos se encuentran correctamente preparados, los archivos de la plataforma descargados, y la configuración de nuestra red LTE perfectamente modificada.

En primer lugar, realizaremos la compilación de los archivos para la puesta en funcionamiento del EPC o CN, a través del terminal:

```
$ ./cmake_targets/build_hss -i
$ ./cmake_targets/build_mme -i
$ ./cmake_targets/build_spgw -i
```

En segundo lugar, lanzaremos el script para la generación de los certificados de cifrado, tanto para el HSS como para el MME:

```
$ ./check_hss_s6a_certificate [ruta destino] [nombre_hss.dominio]
$ ./check_mme_s6a_certificate [ruta destino] [nombre_mme.dominio]
```

En tercer lugar, lanzaremos las tres entidades en diferentes terminales: HSS, MME y SPGW.

```
./run_hss [Option]
./run_mme [Option]
./run_spgw [Option]
```

Las opciones que nos permite agregar al lanzamiento de la entidad HSS son las siguientes:

```
-h                          # Ayuda
-c, –config-file filename   # Archivo de configuración para el HSS
                            # si no quieres usar por defecto
-e, –export-db filename     # Exportar la base de datos a un
                            # archivo
                            # SQL
-D, –daemon                 # Lanzar el demonio
-i, –import-db filename     # Importar el archivo SQL a la base de
                            # datos
-I, –install-hss-files      # Instalación de los archivos de
                            # configuración del HSS
-g, –gdb                    # Lanzar con GDB
-h, –help                   # Muestra esta ayuda
-k, –kill                   # Para la entidad local HSS
```

Las opciones que nos permite agregar al lanzamiento de la entidad MME son las siguientes:

```
-h                                  # Ayuda
-c, –config-file filename           # Archivo de configuración para el HSS
                                    # si no quieres usar por defecto
-D, –daemon                         # Lanzar el demonio
-I, –install-mme-files              # Instalación de los archivos de
                                    # configuración del MME
-g, –gdb                            # Lanzar con GDB
-h, –help                           # Muestra esta ayuda
-k, –kill                           # Para la entidad local MME
-m, –mscgen directory               # Genera los archivos de salida mscgen
                                    # en un directorio
-s, –scenario-player scenario_abs_path  # Lanza el MME con un
                                    # escenario
-v, –verbosity-level                # Nivel de vervosidad (0,1,2)
                                    # 0: ASN1 XER printf off
                                    # 1: ASN1 XER printf on and ASN1 debug
                                    # off
                                    # 2: ASN1 XER printf on and ASN1 debug
                                    # on
```

Las opciones que nos permite agregar para la puesta en marcha de la entidad SPGW son:

```
-h                          # Ayuda
-c, –config-file filename   # Archivo de configuración para el HSS si no quieres usar por defecto
-D, –daemon                 # Lanzar el demonio
-I, –install-mme-files      # Instalación de los archivos de configuración del MME
-g, –gdb                    # Lanzar con GDB
-h, –help                   # Muestra esta ayuda
-k, –kill                   # Para la entidad local MME
-m, –mscgen directory       # Genera los archivos de salida mscgen
                            # en un directorio
-v, –verbosity-level        # Nivel de vervosidad (0,1,2)
                            # 0: ASN1 XER printf off
                            # 1: ASN1 XER printf on and ASN1 debug
                            # off
                            # 2: ASN1 XER printf on and ASN1 debug
                            # on
```

En cuarto lugar, realizaremos el lanzamiento del script del eNodeB. Antes de ello debemos realizar la compilación con el siguiente comando:

```
$ ./cmake_targets/build_oai [Options]
```

El significado de los parámetros adicionales es el siguiente:

```
-h,                         # Ayuda
-c, –clean                  # Elimina todos los archivos de
                            # compilaciones anteriores
-C, –clean-all              # Elimina todos los archivos de
                            # compilaciones e instalaciones
–clean-kernel               # Elimina las características
                            # instaladas en el kernel
-I, –install-external-packages # Instala paquetes externos
–install-optional-packages  # Instala paquetes opcionales
                            # creados
-g, –gdb                    # Lanzar con GDB
–UE,                        # Compila partes específicas del UE
-r,                         # Compila según el Rel.10 y limitado
                            # a la Rel.8
-w,                         # Indica el soporte hardware
                            #que vamos a utilizar
-eNB,                       # Crea el softmodem de LTE
-t,                         # Protocolo de transporte: Ethernet o
                            # ninguno
```

```
–oaisim,                          # Compila el simulador OAISI
–phy_simulators,                  # Compila simuladores de la capa
                                  # física
–core_simulators, -s,             # Lanza una serie de auto-test basados
                                  # en simuladores y varias
                                  # compilaciones
–run-group,                       # Esta opción sólo válida si -s
-V, –vcd                          # Debug en archivos binarios
-x, –xforms                       # Incluye el osciloscopio
–install-system-files             # Instala los archivos requeridos por
                                  # OAI
–noS1,                            # Compila sin el interfaz S1
–verbose-compile,                 # Muestra los detalles de la
                                  # compilación en un archivo
–cflags_proccesor,                # Incluye los CFLAGS del procesador
                                  # si no son detectados correctamente
–build-doxygen,
–disable-deadline,                # Desactiva el programador deadline
–enable-deadline,                 # Activa el programador deadline
–disable-cpu-affinity,
–T-tracer,                        # Activa el T tracer
–disable-hardware-dependency,     # Desactiva la dependencia con el HW
                                  # durante la instalación
```

# Bibliografía


[1] *3GPP TS 36.104 v.10.2.0 Release 10, LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception.*

[2] openLTE An open source 3GPP LTE implementation, 2011. https://sourceforge.net/p/openlte/wiki/Home (Último acceso: 27/06/2017).

[3] Desing and implementation of 4G LTE components - eNodeB and UE on SDR platform using srsLTE. *IEEE 4* (2014), 4.

[4] Protocolos E-UTRAN, 2014. http://www.ipv6go.net/lte/eutran_protocolos.php (Último acceso: 21/06/2017).

[5] Página oficial del Ministerio de energía, turismo y agenda digital, 2016. www.minetad.gob.es (Último acceso: 07/09/2017).

[6] BQ, 2017. https://www.bq.com/es/aquaris-x5plus (Último acceso: 08/07/2017).

[7] Comisión Nacional de los Mercados y la Competencia, 2017. https://www.cnmc.es (Último acceso: 07/09/2017).

[8] Cuadro Nacional de Atribución de Frecuencias, 2017. http://www.minetad.gob.es/telecomunicaciones/Espectro/Paginas/CNAF.aspx (Último acceso: 07/09/2017).

[9] Gemalto GemPC Twin/TR (IDBridge CT30), 2017. http://www.smartcardfocus.us/shop/ilp/id463/gemalto-gempc-twin-tr-idbridge-ct30-/p/index.shtml (Último acceso: 15/07/2017).

[10] Keysight Technologies, 2017. http://www.keysight.com/en/pdx-x201715-pn-N9010A/exa-signal-analyzer-10-hz-to-44-ghz (Último acceso: 20/07/2017).

[11] National Instruments, 2017. http://www.ni.com/es-es/support/model.usrp-2901.html (Último acceso: 07/09/2017).

[12] Página oficial de AMARI LTE 100, 2017. https://www.amarisoft.com (Último acceso: 27/06/2017).

[13] Página oficial de SRS LTE, 2017. http://www.softwareradiosystems.com (Último acceso: 27/06/2017).

[14] SDR Based LTE Workshop - United States Tour, 2017. https://www.nutaq.com/lte-workshop-usa (Último acceso: 20/07/2017).

[15] SYSMOCOM, 2017. http://shop.sysmocom.de/products/sysmousim-sjs1 (Último acceso: 15/07/2017).

[16] Toshiba, 2017. http://www.toshiba.es/discontinued-products/satellite-l850-1jr (Último acceso: 08/07/2017).

[17] Xiaomi, 2017. https://www.xiaomi-shop.es/comprar-xiaomi-mi5-espana.html (Último acceso: 08/07/2017).

[18] Cardona, N., Olmos, J. J., García, M., and Monserrat, J. F. *3GPP LTE: hacia la 4G móvil*, first ed. Marcombo Universitaria, 2011.

[19] Comes, R. A., Álvarez, F. B., Palacio, F. C., Ferre, R. F., Romero, J. P., and Roig, O. S. *LTE: Nuevas Tendencias en Comunicaciones Móviles*, first ed. Fundación Vodafone España, 2010.

[20] Guerra, I. Conceptos básicos de telecos: del GSM al LTE, 2010. https://blog.cnmc.es/2010/05/21/conceptos-basicos-del-telecos-evolucion-de-las-comunicaciones-moviles-del-gsm-al-lte (Último acceso: 03/06/2017).

[21] kumar Swamy Pasupuleti. EMM information, 2012. http://howltestuffworks.blogspot.com (Último acceso: 06/06/2017).

[22] Muñoz, J. S. Z., Muñoz, F. D. C., and Flórez, V. M. Q. Análisis del desempeño a nivel físico del enlace de bajada de la evolución a largo termino (LTE). *GTI 12*, 34 (2014). http://revistas.uis.edu.co/index.php/revistagti/article/view/3847.

[23] Netmanias. LTE EMM Procedure: 1. Initial Attach for Unknown UE, 2017. http://www.netmanias.com/ko/post/techdocs/5320/attach-emm-lte/lte-emm-procedure-1-initial-attach-for-unknown-ue-part-2-call-flow-of-initial-attach (Último acceso: 05/06/2017).

[24] oficial de Tutorials point, P. LTE Network Architecture, 2017. https://www.tutorialspoint.com/lte (Último acceso: 02/07/2017).

[25] Serratusell, J. How to calculate OPC, 2013. http://old.smartjac.biz/products/sim-cards/136-how-to-calculate-opc (Último acceso: 16/07/2017).

[26] Shetty, M. LTE as an all IP architecture, endorsed by many leading vendors, 2011. http://www.kmshetty.com/2011/01/lte-as-all-ip-architectureendorsed-by.html (Último acceso: 22/07/2017).

[27] teltronic a Hytera company. Infraestructura lte enebula, 2017. http://www.teltronic.es/productos/lte/infraestructura-lte-enebula (Último acceso: 23/07/2017).

[28] w3ii. LTE turorial, 2017. http://www.w3ii.com (Último acceso: 07/09/2017).

[29] Yacchirema, C. 4G LTE definición, arquitectura, características, cálculos, simulaciones y seguridad, 2014. http://inalambricas-lte4g.blogspot.com.es/2014/08/arquitectura-lte-la-arquitectura-lte.html (Último acceso: 22/07/2017).

[30] Zheng, Q., Du, H., Li, J., Zhang, W., and Li, Q. Open LTE: an open LTE simulator for mobile video streaming. *Paper presented at the 1-2* (2014), 1–2.