Escuela Técnica Superior de Ingenieros
Industriales y de Telecomunicación

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Estudio de plataformas SDR para LTE-5G

(Study of platforms of SDR for LTE-5G)

Para acceder al Titulo de
Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: XueyanXiang

Septiembre - 2018

Resumen

Según la encuesta realizada por el Instituto Nacional de Estadística (INE) en 2017, el 83,4% de los hogares españoles tienen acceso al internet con un aumento de más de 1.5 % en respeto con el año anterior. Esto refleja claramente que cada vez hay más dependencia con el internet debido a un incremento significante de la necesidad de información. A consecuencia de esto, el estándar 5G está pensado como una nueva tecnología que es capaz de cubrir la necesidad de los clientes, tales como el aumento de velocidad de transmisión de datos como la reducción del tiempo de latencia. Para ello, es fundamental el estudio de la red LTE, ya que servirá de base de 5G.

Por estos motivos, este proyecto se utiliza para entender profundamente la arquitectura de la tecnología LTE. Consta de dos partes: La primera se trata de la puesta en marcha de una red celular LTE de cuarta generación usando el software de código libre OAI (Open AIR Interface), tanto en máquinas virtuales como en escenas reales usando una tarjeta de desarrollo bladeRFx40 como eNB, y por el lado, utilizar un dispositivo móvil como el equipo de usuario para conectarse a la estación base creada. A continuación, se sustituirá el equipo de usuario comercial anterior por uno virtualizado siguiendo las mismas ideas de implementaciones realizadas con el OAI. La segunda parte de este proyecto consiste en el análisis del resultado de las implementaciones de escenarios anteriores. De esta manera, se ha podido hacer una evaluación de esta plataforma y entender más fácilmente el funcionamiento de una red LTE.

Palabras claves:

OAI (OpenAirInterface), 4G, LTE (LongTermEvolution), bladeRF, 5G

Abstract

According to the survey conducted by the National Institute of Statistics (INE) in 2017, 83.4% of Spanish households have access to the Internet with an increase of more than 1.5% in respect to the previous year. This reflects clearly that there is increasing dependence on the internet due to a significant increase in the need of information. As a result of this, the 5G standard is designed as a new technology that is able to meet the needs of customers, such as the increase in data transmission speed as the reduction of latency time. For this, the study of the LTE network is fundamental, since it will serve as the basis for 5G.

For these reasons, this project is used to understand deeply the architecture of LTE technology. It consists of two parts: the first one talks about the implementation of the fourth generation LTE cellular network using the open source software OAI (Open AIR Interface), both in virtual machines and in real scenes using a development card called blade RFx40 as eNB (envolved Base Station), and by the side, use a mobile device as the user equipment to connect to the base station created. Later, the previous commercial user equipment will be replaced by a virtualized one following the same ideas of implementations made with the OAI. The second part of this project consists in the analysis of the results of the implementations of previous scenarios. In this way, it has been possible to evaluate this platform and understand more easily the operation of an LTE network.

Keywords:

OAI (OpenAirInterface), 4G, LTE (LongTermEvolution), bladeRF, 5G

Agradecimientos

Quiero aprovechar la ocasión para agradecer unas cuantas personas:

En primer lugar, a mis padres, que siempre me han apoyado en cualquier decisión que he ido tomando, porque piensan que soy una persona suficientemente madura e independiente como para tomar una decisión de las que no me arrepentiré, ya no decimos que si son buenas o malas.

En segundo lugar, a mi tutor académico, Tomás Fernandez Ibáñez. A pesar de que este proyecto no tiene mucho que ver con el campo en el que suele trabajar él, ha hecho todo lo posible para ayudarme, siempre con mucha atención y amabilidad.

Asimismo, no puedo olvidar en este momento, a la empresa *Iker Ian* por darme esta oportunidad y confiar en mí.

Por el último, a mis compañeros de la empresa *Iker Ian*. Empezando primero por la chica más trabajadora e inteligente que he conocido: Zaloa Fernández. Ha sido muy atenta y paciente conmigo. Además, tengo que darle las gracias a Iñaki Val. A pesar de lo ocupado que está casi siempre, sus correos de: “¿cómo te va todo?” No fallan nunca. Y también, al resto de compañeros, tales como Ander, Marco, Victor, Oscar, Lucía, y etc. Todos me han ayudado de alguna manera y creo que este proyecto no habría sido lo mismo sin ellos.

¡Muchas gracias a todos!

Índice

Lista de figuras

Lista de tablas

Capítulo 1. Introducción

En los años 80, se lanzó la 1ª red de comunicación móvil automatizada comercial. Estaba basada en el estándar AMPS (Advanced Mobile Phone System) y la tecnología que se utilizaba era analógica junto con una multiplexación en FDMA. Este tipo de redes era usado solamente para el envío de paquetes de voz y la transmisión de voz no era nada segura.

Por eso, al principio de los años 90, se desplegaron las primeras redes de GSM (Sistema Global para Comunicaciones Móviles), que facilitaba la transmisión de datos digitales además de de voz. Estaba basado en la tecnología digital.

Debido a la velocidad de transmisión y otros factores, a GSM se le considera como el estándar de la 2ª generación, y es la base de la 3ª generación, teniendo su extensión llamada UMTs, que destaca principalmente por su alta velocidad de transmisión de datos a bajo coste y la utilización de diferentes protocolos de radio (W-CDMA).

De la evolución de 3G a 4G, se han desplegado redes que estaban basadas en estándares mejorando los enlaces de transmisión, tanto el UL (UpLink) como el DL (DownLink) y utilización del sistema MIMO en el release 7 de la tecnología HSPA+.

Al final de 2009, a consecuencia de que las redes UMTS hayan llegado al 85% de los abonados de móviles, 3GPP se lanzó un sistema de estándar nuevo con el objetivo de aumentar la velocidad de transmisión de datos, además de mejorar la experiencia del usuario con total movilidad utilizando el protocolo de internet (IP). A este estándar se le conoce como LTE, que servirá de base para la implementación de la red 4G, y posteriormente para el 5G. La figura 1.1 muestra un esquema simple de la evolución de las redes móviles.

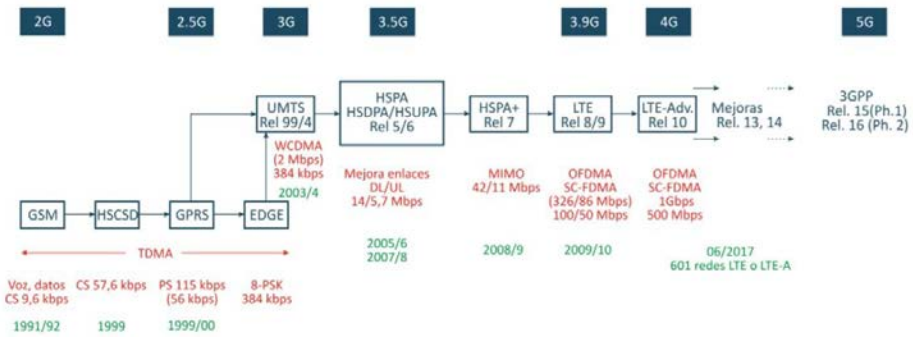

Figura 1. 1 Evolución de redes móviles

Actualmente, se han desplegado 601 redes LTE en 192 países con un porcentaje de más del 50% de la población mundial con cobertura 4G como se muestra en la figura 1.2.

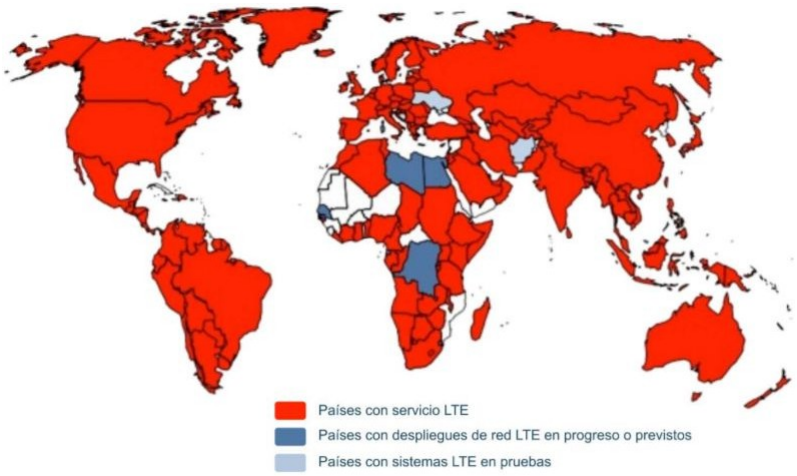

Figura 1. 2 Despliegue mundial de las redes LTE

De esta manera, es algo imprescindible el estudio fundamental de las redes LTE para apoyar a la investigación de 5G, que está en pleno desarrollo.

1.1 Motivación

Como se había comentado anteriormente, debido a la alta demanda del acceso a servicio de Internet por parte de usuarios, es imprescindible la investigación de una tecnología nueva con un tiempo de latencia óptima que será lo que llamado 5G. Aunque hay que destacarse que para su implementación, el 5G necesitará las redes de 4G, como lo que había sucedido en las generaciones anteriores, es decir, cuando comenzó a utilizarse 3G, se necesitó 2G.

Por eso, es importante conocer el funcionamiento de una red celular LTE. En este proyecto se han estudiado los componentes básicos que forman parte de una red LTE. Para ello, se ha desplegado una pequeña red celular propia utilizando la plataforma libre OAI (OpenAirInterface), equipos SDR, en este caso, tarjetas de desarrollo bladeRF, para utilizarlas como la estación base de nuestra red celular LTE, y equipos de PC de altas prestaciones, que es en donde se van a implementar la plataforma. Asimismo, de una manera no costosa, se ha podido estudiar el interior de una red LTE, como el intercambio de la señalización y de tramas.

1.2 Organización del proyecto

El proyecto está organizado en 6 capítulos que se van a dar un breve resumen de cada uno a continuación:

El primer capítulo consiste en una introducción breve al proyecto, entre la cual, se explica la motivación y los objetivos que se quieran conseguir.

El segundo capítulo, se describe el estado de arte de la tecnología LTE y el software OAI para tener un concepto básico de ellos, de esta manera nos serán más fácil entender los resultados.

El tercer capítulo, consta de la implementación de tres diferentes escenarios utilizando OAI. Para ello, se ha hecho primero el escenario virtualizado usando dos máquinas virtuales, y luego, se ha pasado a un escenario de laboratorio, es decir, utilizar tarjetas de desarrollo bladeRF como estación base y un dispositivo móvil como el equipo de usuario. Para el último escenario, se ha sustituido el dispositivo móvil por otra tarjeta de desarrollo con el fin de utilizar ésta como un equipo de usuario no comercial.

El cuarto capítulo, se describe los estudios de los tres escenarios del capítulo anterior. Para ello, se han usado las siguientes herramientas de análisis: wireshark para ver el intercambio de mensajes, el emulador del analizador de tramas T-tracer, el emulador de osciloscopio, un software para el test de velocidad, y etc.

El quinto capítulo, es una estimación de costes para realizar este proyecto, que se divide en los costes de recursos de hardware y de recursos humanos.

Por el último, el capítulo seis se describen las conclusiónes obtenidas, con la finalidad de realizar una exposición breve sobre las hipótesis alcanzadas, opiniones personales, etc.

Capítulo 2. Estado de arte

2.1 Tecnología LTE

Hoy en día l se está investigando el estándar 5G, que consiste en la mejora de la velocidad de subida y de bajada del tráfico en los teléfonos móviles, y la reducción del tiempo de latencia. Actualmente, el estándar considerado como la versión más veloz es el 4G, donde especialmente el estándar LTE, cuyas siglas significan Long Term Evolution (evolución a largo plazo). Este estándar, consiste en una tecnología inalámbrica que se usa para los teléfonos móviles con el objetivo de transmitir datos a alta velocidad.

El estándar LTE es la evolución a largo plazo de 3GPP UMTs; debido al incremento exponencial en el uso de los teléfonos móviles a partir del año 2010, ha surgido la necesidad de lanzar un estándar más potente que el 3G, aumentando la velocidad de las tasas binarias, además de reducir los tiempos de latencia. LTE está basado en 3GPP, pero no llega a ser 4G.

Se resumen las principales características de LTE (release 8) en lo siguiente: [14]

- Uso del espectro: 900 / 1800 / 2000 / 2600 MHz (y otras)
- Flexibilidad de ancho de banda: 1, 4 / 3 / 5 / 10 / 15 / 20 MHz
- Tasas de 100 Mbit/s en DL y 50 Mbit/s en UL
- Retardos reducidos en establecimiento y transmisión (latencia inferior a 10 ms)
- Modos dúplex FDD y TDD
- Multiacceso OFDMA en DL y DFTS-OFDM (SC-FDMA) en UL
- Modulaciones QPSK, 16QAM y 64 QAM
- Técnicas multiantena: diversidad, conformación de haces (beamforming) y MIMO (Mul -tiple Input, Multiple Output)
- Técnicas de protección frente a las variaciones del canal: rate control, channel dependent scheduling, hybridARQ with soft combining
- Coordinación de interferencia (ICIC, Inter—Cell Interference Coordination)
- Compatibilidad con otras tecnologías de 3GPP

2.1.1 Arquitectura del sistema LTE

En este capítulo se va a explicar la arquitectura básica de LTE o Evolved Packet System (EPS). Para ello, se muestra en la siguiente figura un modelo completo de la red LTE a muy alto nivel. Se puede ver que LTE está formado por dos partes, el núcleo de red (CN), EPC, la red de acceso radio (RAN), E-UTRAN y el equipo de usuario (UE). Dentro del núcleo de red, cada nodo engloba, por un lado, mme (Mobility Management Entity), que es el plano de control, cuya principal función es el control de la movilidad de los paquetes, y por el otro lado, S-GW, que es el plano de usuario que se encarga del encaminamiento de datos. Cada nodo de EPC se

conecta directamente con las estaciones bases eNBs de E-UTRAN mediante la interfaz s1, mientras que las estaciones bases eNBs se conectan entre sí a través de la interfaz x2.

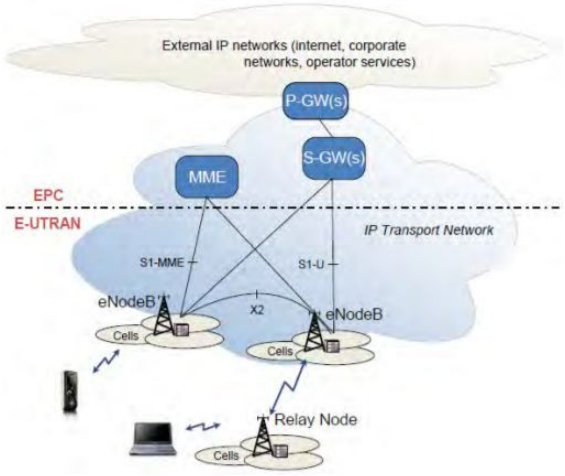

Figura 2. 1 Esquema de la estructura de la tecnología LTE

También hay que destacar que la arquitectura LTE utiliza una tecnología "ALL IP", es decir, se utilizan las tecnologías de red basadas en IP para interconectar diferentes equipos físicos; de esta manera, la red física usada o red de transporte es una red IP convencional. De esta forma, la infraestructura de una red LTE sirve tanto para los equipos propios que realizan funciones del estándar 3GPP como para otros elementos de red propios de las redes IP, tales como routers, servidores DHCP (Dynamic Host Configuration Protocol), y etc.

En la próxima sección, se van explicar detalladamente las arquitecturas EPC, E-UTRAN y UE.

2.1.2 Arquitectura E-UTRAN

Desde los primeros lanzamientos del estándar UMTS, la arquitectura UTRAN inicialmente se asoció fundamentalmente con los conceptos de red de acceso 2G / GSM. La arquitectura general sigue el viejo modelo "estrella" 2G / GSM, lo que significa que con un único controlador (el RNC) se puede controlar un gran número de estaciones base de radio (el Nodo B) sobre la interfaz "Iub". Además, se definió una interfaz "Iur" entre RNC para permitir UTRAN a nivel de RNC y la macro-diversidad entre diferentes nodos B controlados por diferentes RNCs. La macro-diversidad se produjo por las capas físicas UTRAN basadas en CDMA, como un medio para reducir la interferencia de radio y preservar la capacidad de la red. La arquitectura UTRAN inicial dio como resultado una implementación simplificada del Nodo B y un diseño RNC relativamente complejo, sensible, de alta capacidad y completo en funciones. En este modelo, el RNC tuvo que admitir funciones de gestión de recursos y tráfico, así como una parte importante de los protocolos de radio. [11]

Como se había explicado en el apartado anterior, la red de acceso de radio E-UTRAN a diferencia de la arquitectura UTRAN, consta de un único elemento eNB (evolved Node B), éste se encarga de todas las funciones de una red de acceso.

En la figura siguiente se explica la arquitectura de E-UTRAN:

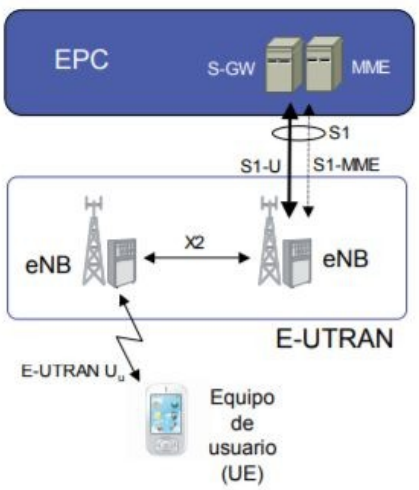

Figura 2. 2 Esquema de estructura de la arquitectura E-UTRAN

Los eNBs son las estaciones bases en E-UTRAN, y tienen la finalidad de conectar la red troncal EPC y los equipos de usuario (UE). Tal y como se muestra en la figura, se aprecian tres interfaces que se utilizan entre las comunicaciones de elementos: S1, X2 y E-UTRAN Uu.

La interfaz S1 se utiliza entre eNB y la red troncal EPC, gracias a ella, se puede conectar directamente eNB a EPC. Se diferencia en dos tipos: S1-U (S1 User Plane) y S1-MME. S1-U se encarga de la función del plano de usuario y se conecta a la entidad S-GW, mientras que S1-MME es el plano de control y es conectado solamente a la entidad MME. El plano de usuario de una interfaz gestiona el transporte de tráfico de usuario de dicha interfaz, y por el otro lado, el plano de control hace que se transmitan los mensajes de control necesarios para el funcionamiento del sistema de la misma interfaz (señalización). Así, se separan las entidades de la red troncal para que los recursos necesarios de transmisión del sistema LTE puedan realizar independientemente la señalización del sistema y el envío de tráfico de los usuarios.

Como se ha explicado previamente, se realiza la transferencia de datos entre eNB y S-GW a través de S1-U, y los datos llegan sin garantía de entrega, no tiene control de errores ni control de flujos, ya que el protocolo que se utiliza para multiplexar los paquetes IP de varios usuarios en esta interfaz S1-U se denominan GTP-U, y se basan en los protocolos UDP/IP.

En cuanto a la interfaz que se utiliza entre eNB y MME, S1-MME, ésta sustenta un conjunto de funciones y controles, entre los cuales, se destacan los siguientes: [6]

- Los procedimientos para establecer, modificar y liberar recursos de servicios incluidos en la interfaz radio y en la interfaz S1.

La red de acceso E-UTRAN ofrece lo que se denomina E-RAB (E-UTRAN Radio Access Bearer), que es la concatenación de servicio portador radio y S1, y ambos forman el servicio portador.
Además, en LTE, es la red troncal el que se encarga del control del plano de usuario, en concreto, lo hace desde la entidad de red MME. Por eso, ni un eNB ni un equipo de usuario puede establecer un servicio portador de radio.

- Procedimientos de handover entre eNBs. Durante una conexión, si la red E-UTRAN quiere que un terminal cambie de eNB y que ante esa situación no haya ninguna interfaz X2 entre los dos eNBs involucrados, el procedimiento de handover se puede realizar a través de la interfaz S1-MME. Asimismo, sin tener que realizar de nuevo todos los procedimientos previos para el establecimiento de los servicios portadores, la entidad MME se establece un nuevo contexto en el eNB destino asociado al terminal, y realiza todos los cambios necesarios.

- Procedimiento de aviso (Paging). La entidad MME sustenta la función de la gestión de la localización de los equipos de usuario en la red. De esta manera, se puede localizar con cierto nivel de resolución a un usuario que esté en modo idle, es decir, cuando éste no tiene ninguna conexión de control RRC establecida con eNB. Para ello, mediante la interfaz S1-MME, se ordenan mecanismos de aviso en todos los eNBs posibles para poder encontrar al terminal, de esta manera, el MME puede cambiar el modo de un usuario de idle a activo.

- Procedimiento de envío de mensajes de señalización y de control que se transportan entre el MME y el equipo de usuario de una forma transparente entre MME y eNB. Se utilizan los protocolos NAS (Non Access Stratrum) para estos mensajes.

A continuación, se va a explicar la interfaz X2, que se utiliza en la comunicación entre los nodos eNB. Su función principal es minimizar la pérdida de paquetes debido a la movilidad de ellos. Tiene dos planos: el plano del usuario, que está basado en GPT-U (GPRS Protocol Tunnel), UDP e IP; y el plano del control que utiliza SCTP e IP. De esta manera, a medida que el terminal se mueve a través de la red de acceso, los paquetes no enviados o no reconocidos almacenados en las colas de los viejos eNBs pueden reenviarse o tunelizarse al nuevo eNB gracias a la interfaz X2. También es importante destacar que debido a la existencia de esta interfaz, se establece un plano de usuario entre eNBs durante la transmisión de datos en el proceso de handover. De esta manera, tanto la señalización de datos como la transferencia de estos puede llevarse a cabo directamente entre eNBs sin tener la necesidad de pasar por el nodo MME de la red troncal EPC. [11]

Por el último, la interfaz E-UTRAN Uu o interfaz radio LTE es la interfaz que se ocupa de la transmisión de datos entre el nodo eNB y los equipos de usuario por vía radio. Se encarga el eNB todos los controles relacionados con la transferencia de datos e implementación de protocolos usados en esta interfaz. [6]

2.1.3 Arquitectura del protocolo de radio de LTE

En este apartado, se van a explicar los diferentes protocolos utilizados en la arquitectura LTE. Para ello, se diferencian en dos tipos: los protocolos del plano de usuario y los protocolos del plano de control:

En la interfaz LTE-Uu, que es la que se encarga de la transferencia de datos entre el nodo eNB y los equipos de usuario, se sustenta en protocolos formados por una capa de enlace de datos y una capa física. En la figura 2.3 se muestra el modelo de la pila de protocolos del plano de usuario para la arquitectura LTE. [7]

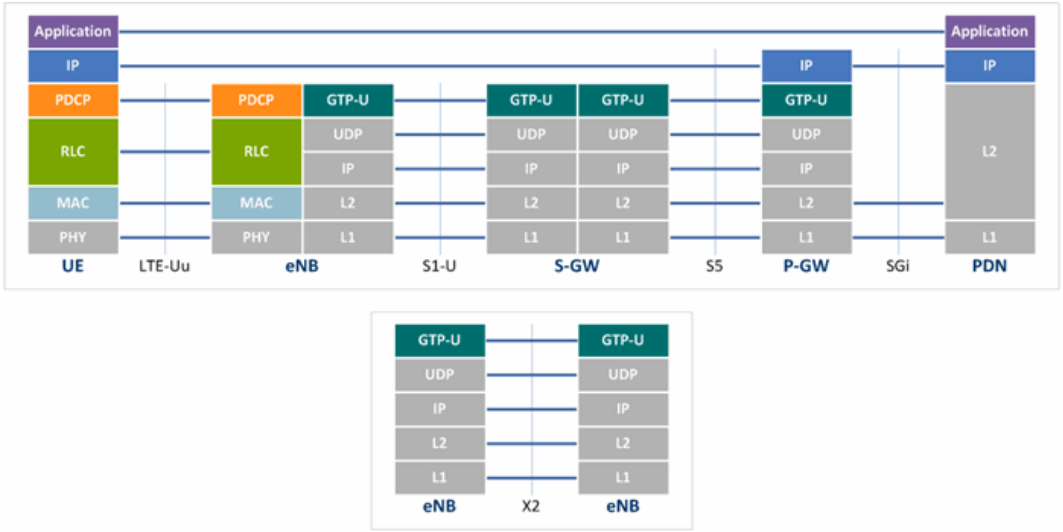

Figura 2. 3 Pila de protocolos del plano de usuario [7]

La capa de enlace de la dicha interfaz se desglosa en tres subcapas: PDCP, RLC y MAC. Cada capa/subcapa se ocupa de un conjunto de funciones concretas y define el formato de los paquetes de datos (e.g., cabeceras y colas) que se intercambian entre las distintas entidades. A continuación, se da una explicación breve de las distintas capas:

- La capa física: la información que procede de los canales de transporte de la capa MAC es transportada a través de esta capa mediante el aire. Algunas de sus funciones son la adaptación del enlace (AMC - Adaptive Modulation and Coding), el control de potencia, la búsqueda de células (para sincronización inicial y handovers) y otras mediciones para la capa RRC. [6]

- La capa PDCP: utiliza el protocolo PDCP para realizar el transporte eficiente de paquetes IP a través del enlace de radio. Las principales funciones son las siguientes:

  - ☐ La compresión/descompresión de encabezado. Para ello, se utiliza el protocolo ROHC (Robust Header Compresion). Además, para poder identificar al paquete IP enviado y garantizar una llegada ordenada de éstos en el lado receptor,

evitando de esta manera los posibles duplicados de los paquetes IP, se añade por esta capa una cabecera que lleva un número de secuencias.

- ☐ El control de seguridad de nivel de acceso (AS) (cifrado y protección de integridad). Los mecanismos de cifrado son obligatorios en los flujos de señalización que se transmiten a través del plano de control, mientas que el cifrado es opcional en los flujos de datos transmitidos a través del plano de usuario
- ☐ El reordenamiento / retransmisión de paquetes durante el traspaso.

- La capa RLC: es el responsable de una transmisión de paquetes por vía aire. En el lado de transmisión, es el protocolo RLC que construye PDU_RLC y proporciona la PDU_RLC a la capa MAC. El protocolo RLC realiza la segmentación y la concatenación de PDCP_PDU durante la construcción de la PDU_RLC. Por el otro lado, en el caso de la recepción, el protocolo RLC realiza el reensamblaje de la PDU_RLC para reconstruir la PDU_PDCP. El protocolo RLC tiene tres modos operativos, es decir, el modo transparente, el modo reconocido, y el modo no reconocido. Y cada uno ofrece diferentes niveles de fiabilidad. A parte de todo esto, la capa RLC también realiza el reordenamiento y la retransmisión de paquetes (PDU_RLC).

- La capa MAC: se encuentra entre la capa RLC y la capa PHY. Por un lado, está conectado a la capa RLC a través de canales lógicos, y por el otro lado, a la capa PHY a través de canales de transporte. Por lo tanto, el protocolo MAC se encarga de la multiplexación y de la demultiplexación entre canales lógicos y canales de transporte. Las capas superiores usan diferentes canales lógicos para diferentes métricas de QoS. El protocolo MAC permite que el QoS pueda programar y priorizar datos de canales lógicos. El planificador de eNB asegura que los recursos de radio sean asignados dinámicamente a los UE y realiza el control de QoS para garantizar que a cada portador se le asigne un QoS correspondiente.

El protocolo GTP-U1 se utiliza para reenviar paquetes IP de usuario a través de las interfaces S1-U, S5 y X2. Cuando se establece un túnel GTP para el reenvío de datos durante el traspaso de LTE, un paquete de marcador final se transfiere como el último paquete sobre el túnel de GTP. [7]

Seguidamente, presentamos en la figura 2.4 el modelo de la pila de protocolos del plano de control:

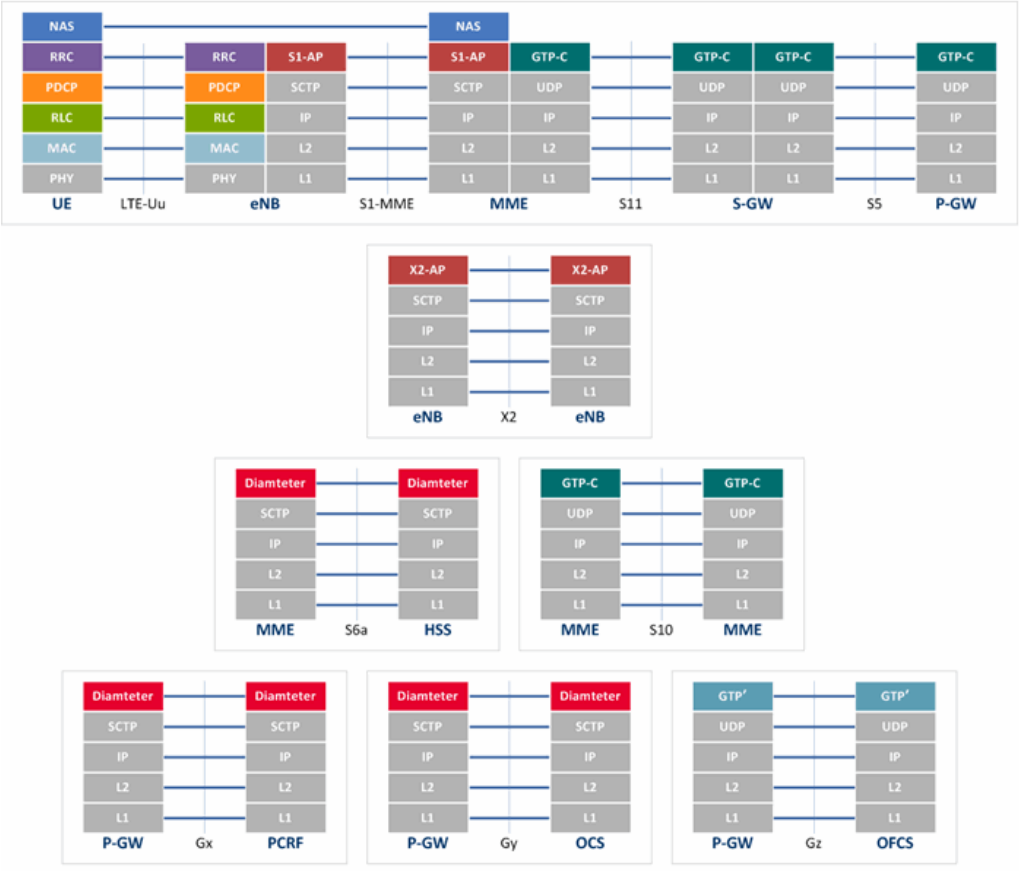

Figura 2. 4 Pila de protocolo del plano de control [7]

Al igual que en el plano de usuario, la capa de enlace de la Interfaz LTE-Uu está formada por la capa PDCP, RLC y MAC. A parte de estas, se añaden otras dos más:

- Capa NAS2: el protocolo NAS realiza funciones de administración de movilidad y administración de portadores.
- Capa RRC: el protocolo RRC permite la transferencia de la señalización NAS. También realiza las funciones necesarias para una gestión eficiente de los recursos de radio. Las principales funciones son las siguientes:

☐ Transmisión de información del sistema
☐ Configuración, reconfiguración, restablecimiento y liberación de la conexión RRC
☐ Configuración, modificación y liberación del portador de radio

En la interfaz X2 se utiliza el protocolo X2AP, que es el que se ocupa de la movilidad del UE y las funciones SON dentro de la arquitectura E-UTRAN. Para soportar la movilidad del UE, el protocolo X2AP sustenta las funciones tales como el reenvío de datos del usuario, la transferencia del estado del SN y la liberación del contexto del UE. En cuanto a las funciones de SON, los eNB intercambian información sobre el estado de recursos, información de carga de tráfico e información de actualización de configuración de eNB, y se coordinan entre sí para ajustar los parámetros de movilidad utilizando dicho protocolo. [7]

Mientras tanto, en la interfaz S1-MME se emplea el protocolo S1AP que se encargan de las funciones como la administración de interfaz S1, la administración de E-RAB, el transporte de señalización NAS y la gestión de contexto de UE. Aparte de eso, también se utiliza para la entrega del contexto de UE inicial al eNB con la finalidad de configurar E-RAB (s) y gestionar la modificación o liberación del contexto de UE a partir de entonces.

Posteriormente, se explican brevemente otros protocolos en función de la interfaz que les corresponde:

- Interfaces S11/S5/S10
  El protocolo GTP-C: GTP-C realiza el intercambio de información de control para creación, modificación y terminación para túneles GTP. Crea túneles de reenvío de datos en caso de transferencia de LTE.

- Interfaz S6a
  Diámetro: el protocolo de diámetro soporta el intercambio de suscripción y la información de autenticación de suscriptor entre HSS y MME.

- Interfaz Gx
  Diámetro: el protocolo de diámetro se responsabiliza de la entrega de reglas de PCC desde la PCRF hasta la PCEF (P-GW).

- Interfaz de Gy
  Diámetro: el protocolo de diámetro se encarga del intercambio de información de control de crédito en tiempo real entre P-GW y OCS.

- Interfaz Gz
  El protocolo GTP ': GTP' realiza la transferencia de CDR del P-GW al OFCS. [7]

2.1.4 Arquitectura de la red troncal EPC

En un principio, se ha diseñado la red troncal EPC para dar un servicio de conectividad IP optimizando la arquitectura de red para poder cubrir las necesidades que requiere la red de acceso E-UTRAN, también se ha tenido en cuenta para el diseño de la red troncal la posibilidad de acceder a sus servicios a través de otras redes de acceso, tales como 3GPP (UTRAN y GERAN) y el ámbito del 3GPP (cdma2000, WiMAX, 802.11).

En la figura 2.5 se muestra la arquitectura de la red troncal EPC formada, por un lado, por las entidades de red que forman su núcleo, y por el otro lado, por las entidades de red e interfaces que se ocupan del control del servicio de la conectividad y de los mecanismos de tarificación. También es importante destacar que una implementación concreta de la red troncal EPC permite que las entidades que tengan distintas funciones puedan habitar en el mismo equipo físico.

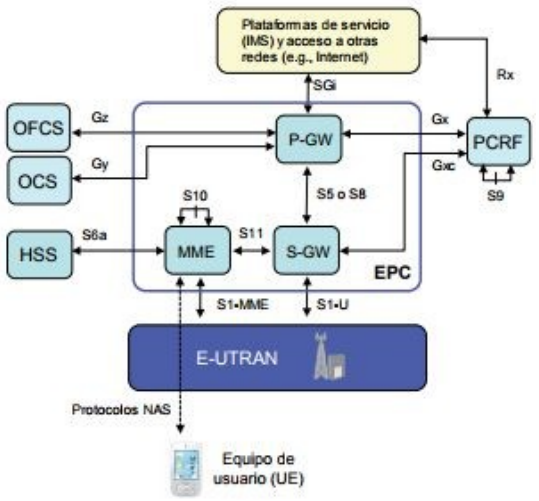

Figura 2. 5 Esquema de estructura de EPC

Como se puede ver en la figura, el núcleo de la red troncal EPC está compuesto principalmente por tres entidades: MME (Mobility Management Entity), Serving Gateway (S-GW) y Packet Data Network Gateway (P-GW). Juntando esas tres entidades con HSS (Home Subscriber Server) que es la base de datos principal del sistema 3GPP se forman los elementos básicos para dar conectividad IP entre los equipos de usuario y redes externas conectadas mediante la EPC. Como se había explicado en los capítulos anteriores, se conecta la red troncal EPC a E-UTRAN mediante la interfaz S1, y entrando más en detalle, entre las entidades de MME, la interconexión se realiza mediante la interfaz S10, mientras que entre una entidad MME y S-GW, se comunica a través de la interfaz S11. Para acceder a información de usuario para poder realizar la autenticación con los equipos de usuario posteriormente, se conecta MME a HSS mediante la interfaz S6a. [6]

Seguidamente, se va a dar una explicación más completa de las entidades que forman parte de la arquitectura de la red troncal EPC:

La entidad MME está a cargo de todas las funciones del plano de control relacionadas con la administración de suscriptores y sesiones. Desde esa perspectiva, MME soporta las siguientes funciones: [11]

- Procedimientos de seguridad: se relaciona con la autenticación del usuario final, así como con la iniciación y negociación de algoritmos de cifrado y protección de la integridad.

- Manejo de sesión de terminal a red: se relaciona con todos los procedimientos de señalización utilizados para configurar el contexto de Datos de paquetes y negociar los parámetros asociados, como la Calidad del servicio.

- Gestión de la ubicación de la terminal inactiva: se utiliza para el proceso de actualización de la zona de seguimiento utilizado para que la red pueda unirse a las terminales en caso de sesiones entrantes.

La entidad HSS (Home Subscriber Server) es la concatenación del HLR (Home Location Register) y el AuC (Authentication Center), son las dos funciones que ya están presentes en las redes pre-IMS 2G / GSM y 3G / UMTS. La parte HLR del HSS se encarga de almacenar y actualizar cuando sea necesario, mientras que la base de datos contiene toda la información de suscripción del usuario, incluyendo los siguientes: [11]

- Identificación y direccionamiento del usuario: corresponde a IMSI (Identidad del Suscriptor Móvil Internacional) y al MSISDN (Número RDSI del Suscriptor Móvil) o al número de teléfono móvil.

- Información de perfil de usuario: incluye los estados de suscripción de servicio y la información de calidad de servicio suscrita por el usuario (como la velocidad de bits máxima permitida o la clase de tráfico permitida).

Por el otro lado, la parte AuC del HSS se ocupa de generar información de seguridad desde las claves de identidad del usuario. Esta información de seguridad se proporciona al HLR y se comunica a otras entidades de la red. La información de seguridad se usa principalmente para la autenticación de terminal de red mutua y la protección de cifrado y la protección de la integridad de la ruta de radio, para garantizar que los datos y la señalización transmitidos entre la red y el terminal no se escuchen ni se alteren.

Desde una perspectiva funcional, S-GW (Serving GW) es el encargado de la terminación de la interfaz de paquetes de datos hacia E-UTRAN. Cuando los terminales se mueven a través de eNodeB en E-UTRAN, Serving GW sirve como el punto de anclaje del plano de usuario en la red troncal, lo que significa que los paquetes se enrutan a través de este punto para movilidad y movilidad interna entre E-UTRAN con otras tecnologías 3GPP, como 2G / GSM y 3G / UMTS.

De forma similar a la entidad S-GW, la puerta de enlace PDN es el punto de terminación de la interfaz de paquetes de datos hacia la Red de datos por paquetes. Siendo el punto de anclaje para la interconexión con las Redes externas de datos, P-GW también soporta funciones de aplicación de políticas (que aplican reglas definidas por el operador para la asignación y uso de recursos) así como el filtrado de paquetes (como inspección profunda de paquetes para detección de firmas de virus) y el soporte de carga evolucionado (como por carga de URL).

Mediante la interfaz Gx con la entidad P-GW se establece la interconexión con la entidad PCRF, se sabe que el servidor de PCRF administra la política de servicio y envía información de configuración de QoS para cada sesión de usuario e información de regla de contabilidad. Además, también combina funcionalidades como la función de decisión de política (PDF) y la función de reglas de carga (CRF) para los nodos UMTS. El PDF es la entidad de red donde se toman las decisiones de política. A medida que se vaya configurando la sesión IMS, se intercambian los requisitos de señalización SIP entre el terminal y el P-CSCF. En algún momento del proceso de establecimiento de la sesión, el PDF recibe esos requisitos del P-CSCF y toma decisiones basadas en las reglas del operador de red, tales como:

- ☐ Permitir o rechazar la solicitud de medios.
- ☐ Uso del contexto PDP nuevo o existente para una solicitud de medios entrantes.
- ☐ Verificar la asignación de nuevos recursos contra el máximo autorizado La función de los CRF es proporcionar reglas de cobro definidas por el operador aplicables a cada flujo de datos de servicio.

Asimismo, el CRF selecciona las reglas de cobro relevantes en función de la información proporcionada por el P-CSCF, como el identificador de aplicación, el tipo de transmisión, la velocidad de datos de la aplicación, etc. [11]

2.1.5 Equipos de usuario (UE)

Para acceder a la red LTE mediante la interfaz radio, se utilizan equipos de usuario (User Equipment, UE). Un equipo de usuario puede ser cualquier terminal utilizado por el usuario para comunicarse, es decir, este terminal puede ser un teléfono móvil, un ordenador portátil con un adaptador de banda ancho móvil o cualquier otro dispositivo. El equipo de usuario se consta de tres partes principalmente: [12]

- Terminal que soporte la tecnología LTE.
- Tarjeta SIM / USIM. Estas tarjetas contienen información básica para la identificación y autentificación del usuario que se conecta a la red LTE. Será explicado con más detalle en el subcapítulo siguiente.
- El equipo de usuario se conecta a la estación base de una red a través de la interfaz radio Uu.

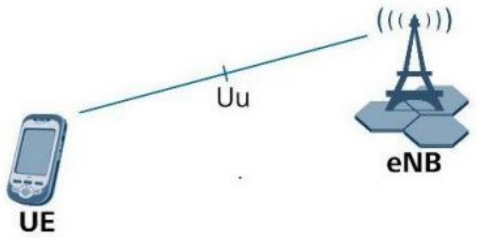

Figura 2. 6 Conexión entre UE y eNB[12]

Las tarjetas USIM

Las redes UMTS (3G) introdujeron la aplicación USIM que se ejecuta sobre un ETSI UICC. La aplicación USIM cubre las partes específicas UMTS, mientras que el ETSI UICC cubre aspectos generales de las tarjetas con chip, independientemente de su aplicación específica.

Un USIM implementa las funciones requeridas por la Autenticación UMTS y el acuerdo clave. Los diferenciadores particulares comparados a la SIM son:

• Autenticación mutua, es decir, el USIM también autentica la red
• Protección de reproducción mediante la introducción de un número de secuencia

La mayoría de los USIM también implementan el protocolo de tarjeta SIM para la compatibilidad con versiones anteriores, por lo que se pueden usar en teléfonos anteriores con solo GSM. [16]

2.1.6 Autenticación

Una vez que el eNB es visible en el equipo de usuario, éste intentará registrarse a él pasando por unos procesos de autenticación para determinar si el UE es un suscriptor autorizado de la red a la que está tratando de acceder. Se explicarán a continuación los procesos de la autenticación.

Código de área de seguimiento (TAC)

Para que el equipo de usuario se conecte correctamente a la red troncal EPC a través de la red E-UTRAN, otro factor importante es el proceso de la autenticación.

En la figura 2.7 se observa la estructura de un Identificador de área de seguimiento (TAI), que está formado por un Código de área de seguimiento (TAC) y una ID de PLMN. Un TAC es el código único que cada operador asigna a cada uno de sus TA. Se denomina una ID de PLMN, a una combinación de un Código de país móvil (MCC) y un Código de red móvil (MNC). Este formato de asignación hace que un TAI sea identificado de manera única a nivel mundial.

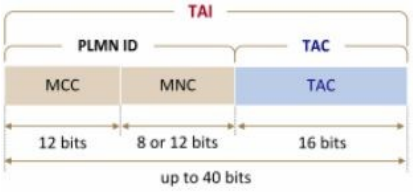

Figura 2. 7 Estructura de un identificador de área de seguimiento (TAI) [10]

La red LTE tiene información de ubicación actualizada acerca de los UE en estado inactivo para descubrir en qué TA está ubicado un UE particular. Para esto, el UE notifica a la red LTE (MME) de su ubicación actual enviando un mensaje TAU (mensaje de solicitud TAU) cada vez que se mueve entre TA. También es interesante destacar que cuando un UE está en estado inactivo envía un mensaje TAU (mensaje de solicitud TAU) a una MME periódicamente incluso cuando el UE permanece dentro de un TA en la lista TAI. [10]

Procedimientos de autenticación

En la figura 2.8 se observan los procedimientos de la autenticación que se van a explicar posteriormente.

El primer paso consiste en el envío *de Attach Request message* en el que el UE solicita acceso a los servicios EPS desde la red MME que sirve. El mensaje incluye las capacidades soportadas por UE y el IMSI de la UE.

Cuando el MME recibe el *Attach Request* provenido del UE, el MME en esta etapa no puede confiar en el UE, por lo que necesita autenticarse para que este UE pueda conectarse a sus servicios. Esto es iniciado enviando un mensaje de *Authentication Information Request* (AIR) al HSS. Este mensaje incluye el IMSI del UE y un identificador para la publicación red.

Al recibir el mensaje de AIR, el HSS comienza a preparar un vector de autenticación (AV) basado en el secreto compartido K que está acoplado con el IMSI. Que consiste en la autenticación esperada respuesta (XRES), una clave de cifrado (CK), una clave de integridad (IK), un desafío aleatorio llamado RAND y un token de autenticación (AUTN). AUT es un parámetro llamado el campo de administración de autenticación (AMF), una autenticación de mensaje código (MAC) y un número de secuencia (SQN) exclusivo O una clave de anonimato (AK).

El MAC está hecho con K, AMF, SQN y RAND. El AK está construido de K y RAND. El XRES también se construye a partir de K y RAND. En la figura 2.9 se observa los distintos componentes de un vector de autenticación. [18]

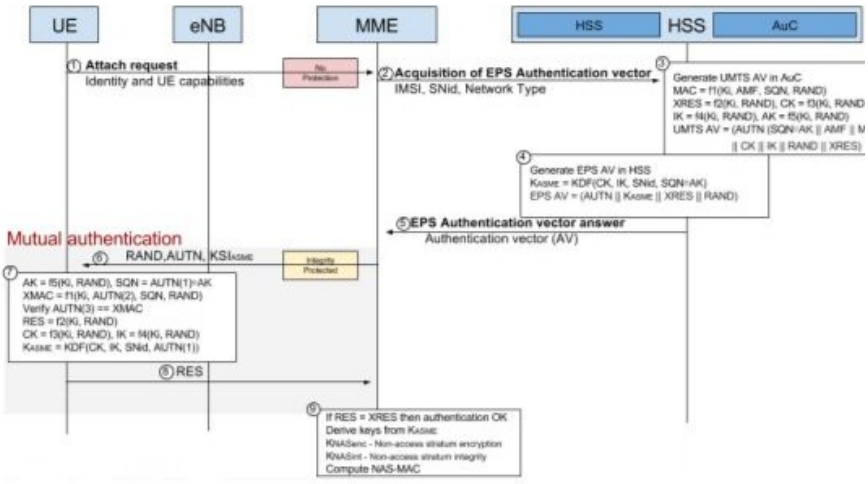

Figura 2. 8 Procedimientos de autenticación [18]

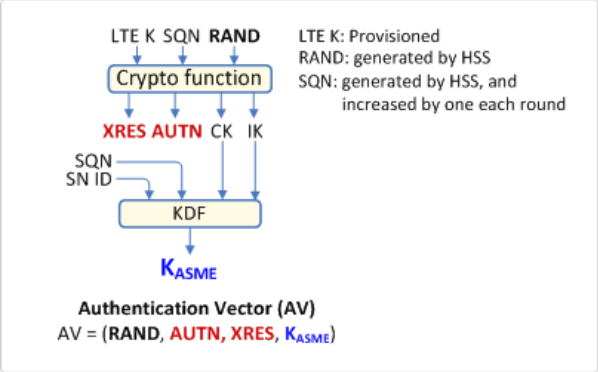

Figura 2. 9 Vector de autenticación [15]

El primer vector de autenticación UMTS AV se emplea para generar EPS AV. Durante este proceso, se genera un nuevo parámetro al utilizar una función de derivación de clave con CK, IK, un identificador de red de servicio (SNID) y SQN exclusive OR AK. El nuevo parámetro es una clave maestra local de 256 bits llamada KASME. Por lo tanto, un vector EPS consiste en AUTN, KASME, XRES yRAND.

Tras la generación del vector de autenticación EPS AV, el HSS le envía un *Authentication Information Answer* (AIA) al MME.

De esta manera, el MME puede enviar el *message Authentication Request* al UE. Lo cual incluye los AUTN y RAND mencionados anteriormente. También contiene un identificador de conjunto de claves, que identifica el conjunto de claves que se utilizará entre el MME y el UE.

Usando el mismo conjunto de algoritmos que el HSS, el UE produce su propia versión de MAC, llamado XMAC. Comparando estos dos valores y si son los mismos, el UE consigue conectarse a la red de servicio. El UE continúa produciendo un parámetro de respuesta llamado RES, que se realiza de la misma manera que XRES. También produce CK e IK y, por último, KASME tal como lo hizo el HSS.

Una vez generados todos los parámetros anteriores, el UE llega a enviar el parámetro RES al MME en *Authentication Response message*.

El MME verifica que el valor recibido de RES sea el mismo que el XRES. Si esto es cierto, entonces la autenticación mutua entre el UE y la red ha sido lograda. Se ha establecido una clave maestra local de sesión compartida, KASME, entre el UE y el MME. Esta clave ahora se puede usar para generar claves de integridad y claves de cifrado para el NAS y el AS. [18]

2.1.7 Estructura de tramas

En la capa física de una tecnología LTE, lo que se conoce como el mínimo elemento de información, es decir, un PRB, es la asignación que realiza un eNB a un UE. Un bloque PRB está compuesto por 12 subportadoras donde se pueden llegar a transmitir hasta 7 símbolos OFDMA y su duración es de 0.5 ms, que es equivalente a un slot o ranura de tiempo. Los datos

que llegan a la capa física de la interfaz radio tienen una forma de bloques de transporte (transport block - TB) con un tamaño variable de duración 1 ms, es decir, dos tiempos de slot.

Cuando se envía un grupo de PRBs que tiene una modulación/codificación común a un usuario en un TTI, a estos se les denominan TB, cuyo tamaño es asignado según el acuerdo llegado por el UE (en el caso del DL) y las medidas de canal que proporciona el eNB (en el UL).

Como se muestra en la figura 2.10 cada subtrama de radio tiene una duración de 1 ms (duración de un TTI) y, por lo tanto, si cada trama tiene 10 subtramas, equivaldría a 10 ms. [6]

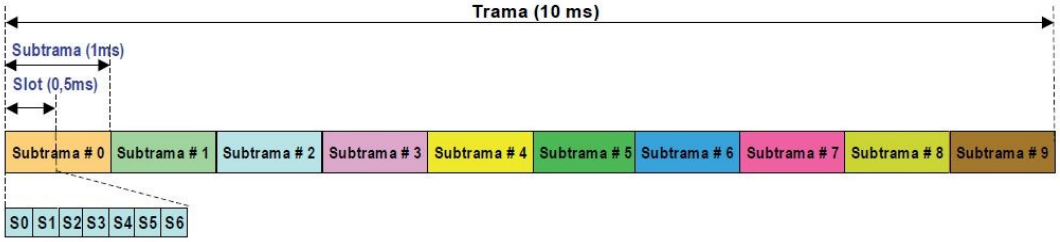

Figura 2. 10 Estructura de la trama de LTE (capa física y FDD)[6]

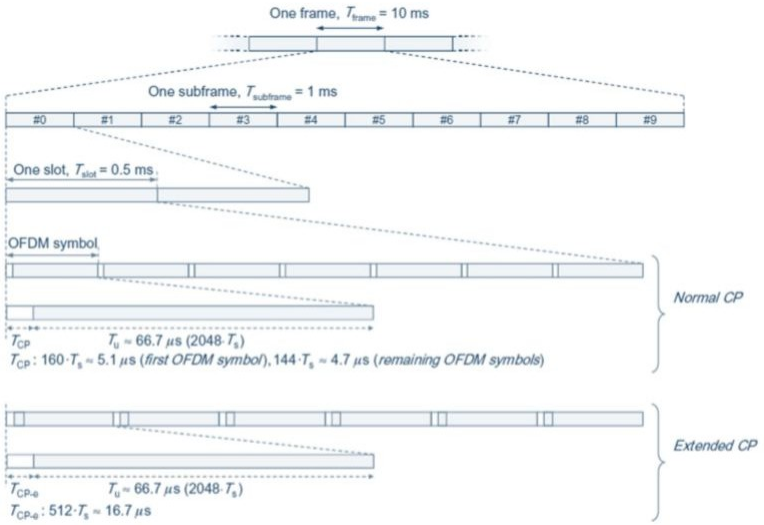

Figura 2. 11 Estructura temporal de LTE [14]

En OFDM, para evitar la interferencia inter-símbolo y poder preservar la ortogonalidad entre las portadoras, se utiliza un tiempo de guarda entre los símbolos. Evitando de esta manera que la cola de un símbolo se solape con el siguiente. Este tiempo de guarda se aprovecha para transmitir el prefijo cíclico, que se diferencia en dos tipos según el número de símbolos que son introducidos en cada intervalo. Si es un prefijo cíclico normal, se añaden 7 símbolos por

intervalo, mientras que, en el prefijo cíclico extendido, se añaden 6, teniendo estos menos símbolos añadidos que el anterior, es menos eficiente, pero es más útil en casos como los escenarios con elevada dispersión del retardo (células extensas).

Dependiendo del tipo de prefijo cíclico que se añada al intervalo, un PRB podría tener tanto 84 como 72 elementos de recursos (RE, ResourceElement). El elemento de recurso consiste en un número de subportadoras moduladas con M niveles (M=4, 16, 64 según sea QPSK, 16QAM o 64 QAM) en un símbolo OFDM como se muestra en la figura 2.12. [14]

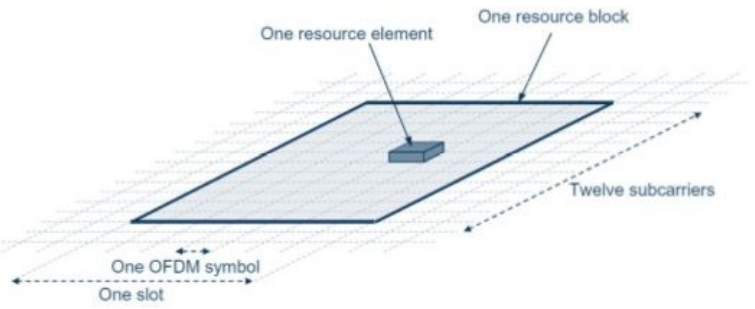

Figura 2. 12 Recurso básico tiempo-frecuencia [14]

A consecuencia de las especificaciones de la capa física que se plasman en la tabla…, puede utilizarse la utilización de cualquier número de RB entre 6 y 110.

La tabla 2.1 muestra los distintos valores que tiene el ancho de banda que se usa en la tecnología LTE en relación con el número de RB y el número de subportadoras. Normalmente, el ancho de banda nominal es mayor que el ancho de banda ocupado en transmisión, dejando de esta manera margen para los límites de espectro.

Ancho de banda nominal(MHz)	1.4	3	5	10	15	20
Ancho de banda ocupado en transmisión(MHz)	1.08	2.7	4.5	9	13.5	18
Número de RB(UL o DL)	6	15	25	50	75	100
Número de subportadoras	72	180	300	600	900	1200

Tabla 2. 1 Parámetros de los diferentes anchos de banda en LTE

2.1.8 Modulación y demodulación

Se destaca que en LTE, se utilizan dos tipos de modulaciones dependiendo del enlace, es decir, si el acceso es en el DL, se utiliza la técnica de acceso múltiple por división de frecuencias ortogonales (OFDMA), en cambio, en acceso del enlace UL, se realiza mediante la técnica del acceso múltiple por división de frecuencia con una única portadora (SC-FDMA).

OFDM

La modulación OFDM está implementada basándose en el procesado IFFT/FFT. En la primera aproximación, el ancho de banda de la modulación se puede calcular según la siguiente expresión:

$$B = N_c * \Delta f \quad \textbf{(Ecuación 1)}$$

Donde Nc es el número de subportadoras, mientras que Δf es la separación entre la frecuencia central de las subportadoras, suele ser 15 KHz con una duración de símbolo de 66.67 µs y un intervalo de guarda de 5 µs.

Se denomina OFDMA (Orthogonal Frequency-Division Multiple Access) al esquema de multiacceso de OFDM que se caracteriza por la asignación de un subconjunto de portadoras a cada usuario de forma fija. En cada intervalo de símbolo OFDM, se utilizan distintos conjuntos de subportadoras para transmitir hacia los distintos UE. Uno de los inconvenientes es el incremento de la diversidad en frecuencia debido al acceso múltiple distribuido. [14]

En cuanto a las modulaciones empleadas en este enlace DL, se pueden utilizar QPSK, 16QAM y 64 QAM dependiendo de número de bits por símbolo.

SC-FDMA

La modulación SC-FDMA utiliza un esquema de transmisión similar a la de OFDMA, diferenciándose en la utilización de una precodificación de los símbolos a transmitir previa al proceso de transmisión OFDM. De esta manera, consigue mejorar en el enlace ascendente (UL) los siguientes parámetros: [6]

- Reducción de la potencia instantánea de la señal transmitida, asimismo, consigue mejorar la eficiencia de los amplificadores de potencia y una reducción de coste.
- Reducción de complejidad de la ecualización en el dominio frecuencial.
- Mayor flexibilidad de la asignación de banda.

2.2 OAI (OpenAirInterface)

En este capítulo se describe el software OAI (OpenAirInterface) que se ha utilizado para la implementación de los distintos escenarios que se han llevado a cabo.

OAI es un software de tecnología inalámbrica de hardware y software de código abierto (simulación, emulación y tiempo real) para el despliegue de redes simuladas con alto nivel de realismo. No requiere ningún hardware de radio para funcionar, ya que puede simular la interfaz de radio utilizada en EPS por Ethernet.
Se ha decantado a utilizar esta plataforma porque OAI proporciona una implementación LTE experimental completa (Rel 8, rel parcial 10) en tiempo real y con funciones bastante completas. Ya que ofrece una implementación de fuente abierta de la interfaz de radio EPS, EUTRAN, y la red central. También está incluida la integridad NAS y la encriptación usando AES, que es un algoritmo de cifrado popular. Además, actualmente, el proyecto está trabajando estrechamente con la comunidad para evolucionar el software hacia futuros lanzamientos 5G de 3GPP. Que es la otra razón por la que nos interesó este software.

OAI se divide en dos partes: Openair-cn y Openairinterface5G que se explicarán detalladamente en los siguientes apartados. A continuación, se muestra en la figura 2.13 un esquema detallado de dónde se encuentran los principales archivos de configuración, ejecución y compilación del sistema.

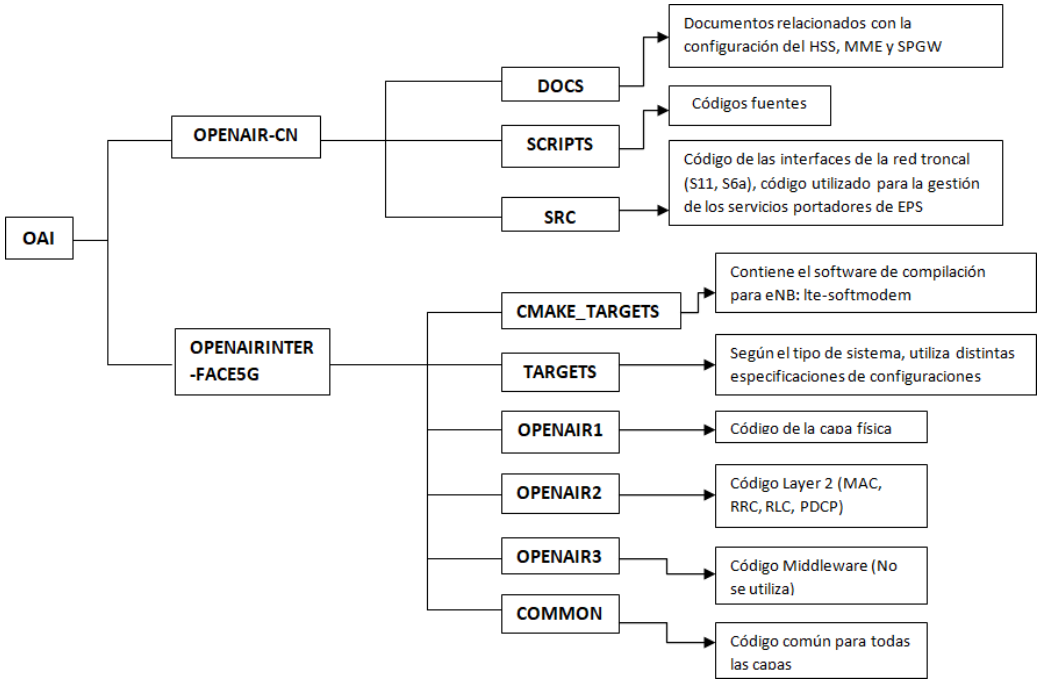

Figura 2. 13 Archivos de configuración de OAI [14]

2.2.1 Openair-cn

Es un software de código libre proporcionado por OSA que se encarga de la implementación de los distintos equipos de la red troncal EPC. Openair-cn consta de las entidades MME y HSS y la pila de protocolos usada para ellos. En particular, el MME y el HSS se ejecutan como propios procesos y cada protocolo o procedimiento principal se ejecuta como hilos en estos procesos. Los hilos se comunican entre sí dentro de un MME o un HSS mediante el uso de la interfaz.

HSS

El proceso HSS consiste en un hilo S6a y una base de datos. La aplicación de base de datos utilizado es MySQL: representa el AuC y almacena los parámetros de autenticación de usuario. El HSS también incluye una base de códigos de autenticación y derivación para el KDF y la función f0. Las funciones criptográficas de nivel inferior, como un HMAC utilizado en el sistema, proviene de una biblioteca de código abierto llamada Nettle, cual es incluido en el HSS.Una implementación de código abierto de Diameter llamada freeDiameter se utiliza para la señalización S6a durante la construcción del HSS y el MME, el diámetro libre se obtiene el código fuente y luego se parchea con el fin de introducir una extensión S6a. [18]

MME

El proceso MME ejecuta varios hilos, donde los de mayor interés son los del protocolo NAS y el hilo S6a. Al igual que con el HSS, el hilo S6a usa freeDiameter para enviar y recibir mensajes en la interfaz S6A. [18]

2.2.2 Openairinterface5G

Es responsable de la simulación eNodeB y UE. Consiste en una combinación abstraída de UE y eNodeB que proporciona datos realistas de la señalización de la pila de radio y del protocolo NAS cuando se conecta al Openair-cn.

2.2.3 Requerimientos de Hardware

Está pensado para el correcto funcionamiento de las siguientes plataformas SDR:

- SDR platform
- ExpressMIMO2
- USRP B2x0, X300
- Blade RF
- LMS-SDR
- Sidekiq (experimental)

En este proyecto, se ha utilizado la plataforma BladeRFx40, en la tabla siguiente se presenta una comparación de dicha plataforma junta a las otras:

	USRP B210	USRP X310	ExpressMIMO2	BladeRF	LMS SDR
Data acquisition	USB3	Gbit Ethernet, PCIexpress	PCIexpress	USB3	USB3
MIMO and bandwidth capabilities	2x1 MIMO 20MHz or 2x2 MIMO 10MHz	2x2 MIMO, 120MHz	4x4 MIMO 5MHz, 2x2 MIMO 10MHz, SISO 20MHz	1x1 SISO 2OMHz	2x2 MIMO 20MHz
RF chip	AD9361	n/a	LMS6002D(x4)	LMS6002D	LMS7002M
Frequency range	70MHz-6GHz	DC-6GHz	300MHz-3.8GHz	300MHz–3.8GHz	300MHz-3.8GHz
Price	€1.130	~€5.000	~€3.000	$420	$299
Duplexing	FDD or TDD	FDD or TDD	FDD or TDD	FDD	FDD or TDD
Output power	10 dBm	10 dBm	0 dBm @2.6GHz 10 dBm @700MHz	6dBm	10dBm

Tabla 2. 2 Especificaciones de las distintas plataformas SDR[9]

Se ha optado por utilizar esta plataforma por su asequible precio, además se adapta perfectamente a la banda que se ha estado trabajando, es decir, a la banda de 2.68 GHz.

También se destaca el necesario uso de un ordenador potente con los básicos requerimientos de computación que se citan a continuación:

- Intel Core i5, i7
- Intel Xeon
- Intel Atom
- 4 cores, > 3GHz, SSE 4, AVX

2.2.4 Implementación de posibles escenarios

El software de OAI ofrece una amplia variedad de posibles implementaciones de escenarios. Se muestra a continuación las distintas configuraciones dependiendo de tipo de componente que sea:

- OAI EPC + OAI eNB + OAI UE
- OAI EPC + OAISIM
- 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 este proyecto, han sido montados los tres primeros escenarios que se explican en el siguiente capítulo.

Capítulo 3. Implementación de escenarios usando OAI

En este capítulo, se han montado tres tipos de escenarios como se había comentado en el capítulo 2.

3.1 Configuración de la red

Para que se puedan conectar el equipo que se encarga de la red troncal con el equipo de E-UTRAN, es necesario configurar las tarjetas de la red IP que se vayan a utilizar; esto se ha hecho después de tener la instalación de equipos, ya que una vez que tengan configuradas las tarjetas de la red IP, los equipos perderían el acceso al internet.

3.1.1 Máquinas virtuales

En la simulación de OAI EPC + OAISIM, se ha usado un ordenador de la marca DELL Latitude E5570, que en él se han creado dos máquinas virtuales usando el software gratuito VMware Workstation 12 Player, ya que es un ordenador suficientemente potente para cumplir los requisitos en este caso. Los recursos que se han utilizado en estas dos máquinas virtuales son lo siguiente:

- Núcleos utilizados: 4
- Memoria RAM: 3 GB
- Memoria HDD: 30 GB
- Tarjetas de red: 2

Se han instalado en estas máquinas el Linux de la versión Ubuntu 16.04. LTS (64 bits), puesto que es un sistema operativo gratuito y preparado para OpenAirInterface. Tras esto, se ha llamado a una de las máquinas virtuales: EPC y la otra OAISIM, y se van a realizar la instalación y la configuración de cada una independientemente.

Antes de empezar con la instalación de los repositorios, lo que se ha llevado a cabo es la configuración de la red entre estas dos máquinas. Para la máquina EPC, se ha usado una tarjeta de red para la conexión a internet, y otra tarjeta para la conexión interna con OAISIM, mientras que en OAISIM, solo se ha usado una única tarjeta de red que servirá de conexión interna con EPC. A continuación, se detalla la configuración de las tarjetas de red en ambas máquinas:

Máquina EPC	
Red para el acceso a internet:	ens33
Dirección IP estática:	192.168.174.135
Máscara de red :	/24
Dirección IP broadcast :	192.168.174.255
Red para la conexión con OAISIM:	ens34
Dirección IP estática:	192.168.1.1
Máscara de red:	/24

Dirección IP broadcast:	192.168.1.255

Tabla 3. 1 Configuración de las tarjetas de red de la MV.EPC

Máquina OAISIM	
Red para la conexión con EPC:	ens34
Dirección IP estática:	192.168.1.2
Máscara de red :	/24
Dirección IP broadcast :	192.168.1.255

Tabla 3. 2 Configuración de las tarjetas de red de la MV.OAISIM

Una vez que se hayan configurado las tarjetas de red correctamente, si se hace un ping desde una máquina a la otra, se podría observar el resultado como los que se muestran en las siguientes figuras:

```
ikerlan@ubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.368 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.194 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.253 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.423 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=0.209 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=0.194 ms
```

Figura 3. 1 Ping desde MV EPC a MV OAISIM

```
ikerlan@ubuntu:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.389 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.452 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.241 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.323 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.235 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=0.235 ms
^C
--- 192.168.1.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5101ms
rtt min/avg/max/mdev = 0.235/0.312/0.452/0.086 ms
```

Figura 3. 2 Ping desde MV OAISIM a MV EPC

Por el otro lado, la máquina EPC tendría que ser capaz de tener acceso a Internet, que se podría comprobar de la siguiente manera:

```
ikerlan@ubuntu:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=22.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=20.9 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=128 time=21.1 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=128 time=25.7 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 20.962/21.951/25.786/1.658 ms
```

Figura 3. 3 Comprobación al acceso a internet

El esquema de la conexión entre las dos máquinas virtuales se muestra en la siguiente figura:

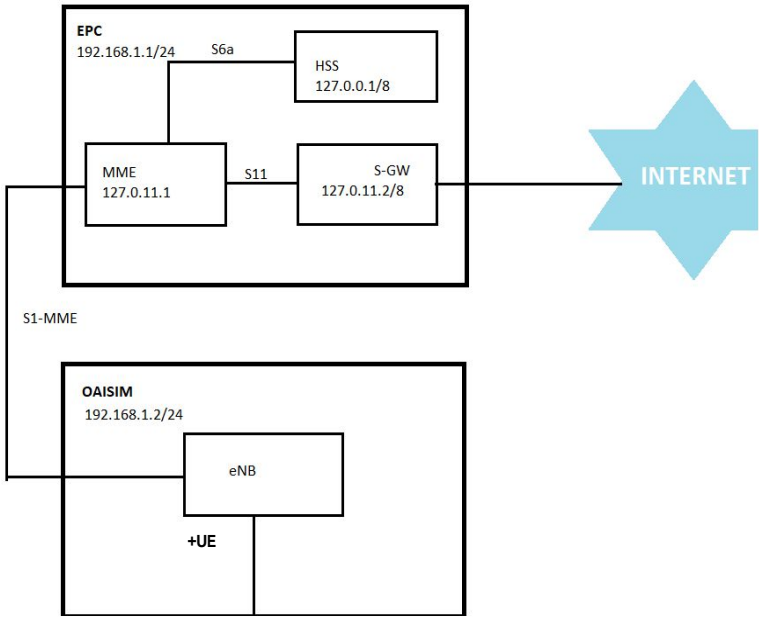

Figura 3. 4 Esquema de conexión entre las MVs

3.1.2 OAI EPC + OAI eNB + UE(comercial)

A diferencia del caso anterior, este es un escenario del laboratorio en el que se ha usado una tarjeta de desarrollo bladeRFx40 como la estación base conectada, por un lado, a un ordenador que se encargará la función de eNB, y por el otro lado, a dos antenas. Para la implementación de este escenario, como se requieren unas especificaciones mucho mayores que el escenario virtual, se han usado dos ordenadores en vez de las máquinas virtuales. Uno es EPC mientras que el otro de eNB. Ambos equipos tienen instalados el sistema operativo Linux de la versión Ubuntu 16.04. LTS (64 bits). A continuación, se muestra en la siguiente figura la implementación de este escenario:

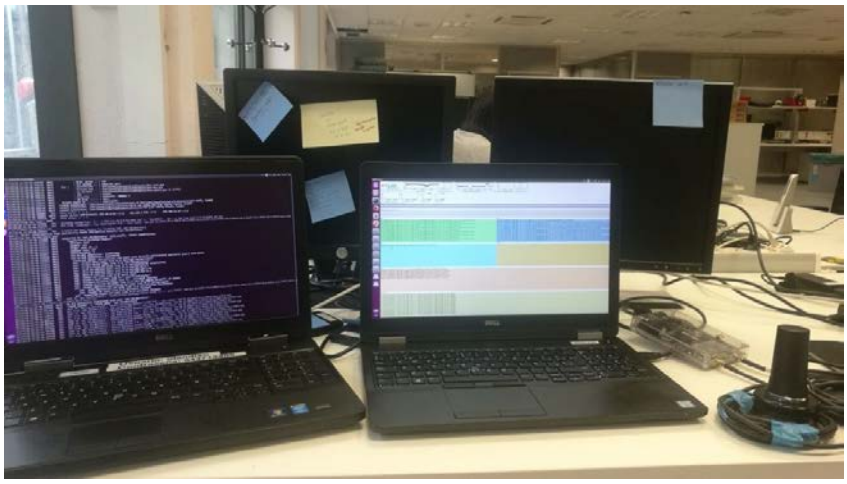

Figura 3. 5 Implementación de escenario de laboratorio

En cuanto a la configuración de las tarjetas de red de estos dos equipos, se destaca que es distinta al caso anterior, y se detalla en la figura siguiente:

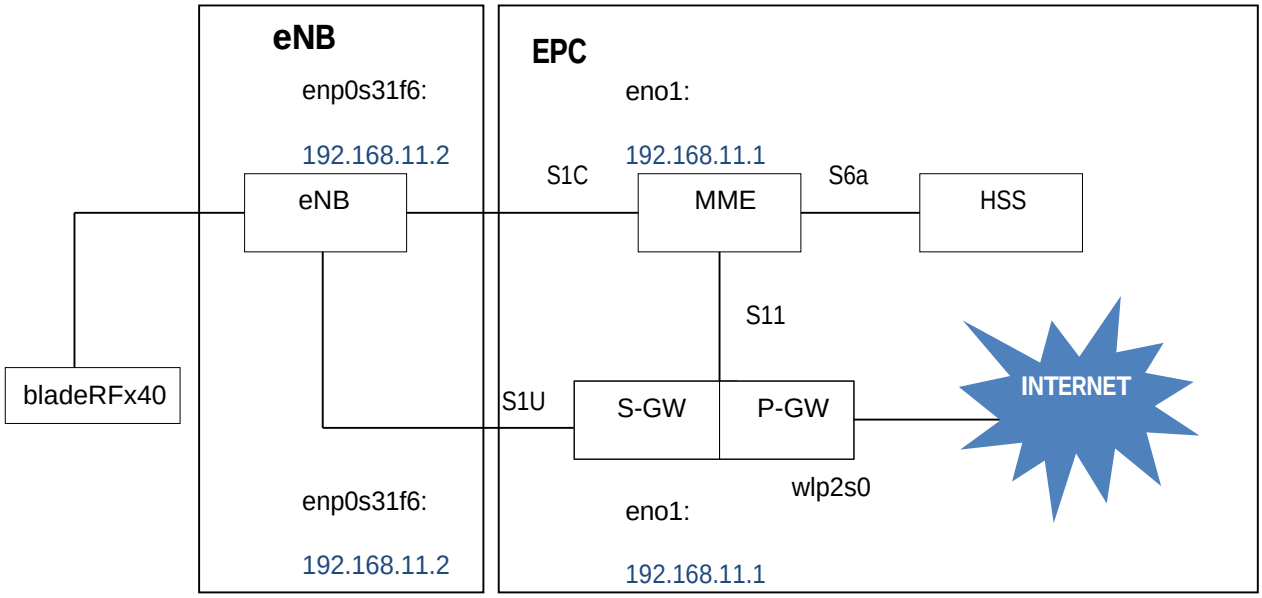

Figura 3. 6 Esquema de conexión en escenario de laboratorio de la red LTE

Como se observa en la figura 3.6, el equipo EPC utiliza una tarjeta de red que tiene una conexión interna con la de eNB mediante la interfaz s1. Para ello, se ha usado un cable conectado entre estos dos equipos. Por el otro lado, el equipo EPC tiene acceso a Internet mediante wifi “wlp2s0”. Se detalla la configuración de las tarjetas de red en las siguientes tablas:

Máquina EPC	
Red para el acceso a internet:	wlp2s0
Dirección IP estática:	-
Máscara de red :	/24
Dirección IP broadcast :	-
Red para la conexión con eNB:	enp0s31f6
Dirección IP estática:	192.168.11.1
Máscara de red:	/24
Dirección IP broadcast:	192.168.11.255

Tabla 3. 3 Configuración de la tarjeta de red de EPC en el escenario de laboratorio

Equipo eNB	
Red para la conexión con EPC:	Enp0s31f6
Dirección IP estática:	192.168.11.2
Máscara de red :	/24
Dirección IP broadcast :	192.168.11.255

Tabla 3. 4 Configuración de la tarjeta de red de eNB en el escenario de laboratorio

3.2 Preparación de las máquinas

Para empezar, para evitar que pida la contraseña cada vez que se ejecute como el administrador, se añade la siguiente línea en el archivo "/etc/sudoers":

```
xxxxxx ALL=(ALL) NOPASSWD: ALL (xxxxxx es el nombre del usuario correspondiente en nuestro equipo)
```

Se ha trabajado con dos escenarios, uno virtual y otro del laboratorio. En ambos casos, se ha instalado el Ubuntu de versión 16.04.1 TLS en cada máquina. En el primer caso, se ha usado solamente un ordenador con dos máquinas virtuales de VM que han sido explicados detalladamente en el apartado 3.1.

Para el escenario del laboratorio, es decir, OAI EPC + OAI eNB + UE comercial, se han utilizado dos ordenadores, ya que para la correcta conexión con el UE comercial, se requiere un rendimiento muy alto del equipo que no ofrecen las máquinas virtuales. Cada ordenador se encargará de una función distinta, uno de ellos desempeñará la función de EPC, en el cual estarán incluidos el MME, el HSS y el SPGW, mientras que el otro servirá del eNB. En el escenario virtual, se encargará cada máquina virtual de la función de los ordenadores.

Con el siguiente comando se ha instalado el Kernel 4.8.0-58 de baja latencia para OAI para ambos ordenadores:

```
sudo apt-get install linux-image-4.8.0-58-lowlatency linux-headers-4.8.0-58-lowlatency
```

Con el comando:

```
uname -a
```

Se comprueba la correcta instalación del kernel.

A continuación, se han desactivado el control de frecuencia de la CPU, los estados C, los estados P y cualquier otra administración de energía del BIOS. Para ello, han sido realizados los siguientes pasos:

En el archivo /etc/default/grub se ha añadido la siguiente línea:

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

Se han actualizado los cambios previos con el comando update-grub .

Tras esto, en el archivo /etc/modprobe.d/blacklist.conf , se ha añadido la siguiente línea al final del todo para meter powerclamp en la lista negra:

```
blacklist intel_powerclamp
```

Se ha instalado la herramienta cpufrequtils con el fin de desactivar el escalado en frecuencia:

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

Se creyó el archivo "cpufrequtils" en el directorio /etc/default, y ha sido añadida la siguiente línea:

```
GOVERNOR="performance"
```

Por el último, para evitar que se sobreescriban los ajustes realizados al reiniciar el equipo, se ha desactivado el demonio "Ondemand" de la siguiente manera:

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

Antes de comenzar con la instalación de OAI, se ha instalado en ambos equipos con el siguiente comando la nueva versión del git para poder descargar los softwares que se vayan a utilizar:

```
sudo apt-get update

sudo apt-get install subversion git
```

Una vez hecho correctamente los pasos del apartado 3.2, se ha procedido a la instalación y configuración de OAI EPC y eNB.

3.3 Instalación de EPC

En primer lugar, se han descargado los códigos fuente con el siguiente comando:

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

En segundo lugar, se han aplicado unos "patches" de EPC siguiendo las instrucciones siguientes para solucionar los errores que podrían tener esta versión de EPC.

```
Patches

cd openair-cn

git checkout develop
```

Se ha aplicado el patch al equipo EPC:

```
git apply ~/opencells-mods/EPC.patch
```

Para continuar, se debería dar un nombre de dominio llamado: Fully Qualified Domain Name (FQDN) a este equipo EPC. Para ello, en el archivo /etc/hosts se ha añadido lo siguiente:

```
127.0.0.1 localhost

127.0.1.1  epc.5GLaboratory epc

127.0.1.2  hss.5GLaboratory hss
```

A continuación se utiliza el comando `hostname -f` para comprobar que la configuración previa es correcta.

En este caso el resultado obtenido es epc.5GLaboratory.

Finalmente antes de realizar la instalación, se ha creado una carpeta en /usr/local/etc/ llamada oai, en la cual se guardarán una copia de archivos de configuración de EPC que posteriormente serán editadas directamente en ellos:

```
sudo mkdir -p /usr/local/etc/oai/freeDiameter

sudo cp ~/openair-cn/ETC/mme.conf /usr/local/etc/oai

sudo cp ~/openair-cn/ETC/hss.conf /usr/local/etc/oai

sudo cp ~/openair-cn/ETC/spgw.conf /usr/local/etc/oai

sudo cp ~/openair-cn/ETC/acl.conf /usr/local/etc/oai/freeDiameter

sudo cp ~/openair-cn/ETC/mme_fd.conf /usr/local/etc/oai/freeDiameter

sudo cp ~/openair-cn/ETC/hss_fd.conf /usr/local/etc/oai/freeDiameter
```

Tras esto, se ha tenido que ejecutar los siguientes scripts, que están situados en la carpeta /openair-cn/SCRIPTS:

```
./build_hss -i

./build_mme –i

./build_spgw –i
```

Con la instalación de hss, se instala automáticamente la herramienta mysql, que con ésta, se podrá hacer modificaciones en la base de datos posteriormente. La opción -i se utiliza solamente para la instalación de éstos, por lo tanto, solo es necesario ejecutarlo la primera vez.

3.4 Configuración del EPC

Para hacer uso de la base de datos, se ha optado por acceder a ella mediante phpmyadmin, que es una plataforma de software libre para poder administrar MySQL vía web, basta con poner la dirección IP de la instancia del servidor al navegador como se muestra a continuación: 127.0.0.1/phpmyadmin. En nuestro caso como no ha sido encontrada la página deseada, para solucionar este problema, es necesario añadir phpmyadmin a la configuración de Apache de la siguiente manera:

Se ha accedido al archivo apache2.conf desde una terminal:

```
vim /etc/apache2/apache2.conf
```

Se ha añadido el siguiente comando al archivo correspondiente:

```
Include /etc/phpmyadmin/apache2.conf
```

Para que los cambios se hagan efecto, se ha tenido que reiniciar apache:

```
/etc/init.d/apache2 restart
```

También es importante añadir los nuevos MCC y MNC(en este caso, es [901,70]) que se han utilizado para poder pasar por la fase de autenticación de la tarjeta USIM en el archivo mcc_mnc_itu.c encontrado en el directorio "/openair-cn/SRC/UTILS/".

3.4.1 Configuración de la base de datos

Como había sido comentado anteriormente, desde phpmyadmin se ha accedido a la configuración de nuestra base de datos, y se ha importado la base de datos llamada oai_db.sql, encontrada en la ruta /openair-cn/SRC/OAI HSS/db/ para el uso de OAI.

Una vez hecho esto, se ha realizado ciertas modificaciones en las tablas que se encuentran dentro de la base de datos importada dependiendo del escenario en el que se ha estado trabajando de la siguiente manera:

En EPC + OAISIM

Siendo un escenario virtual, solo es necesario hacer modificaciones en la tabla mmeidentity, añadiendo en ella el nombre de los equipos involucrados y sus respectivos dominios:

idmmeidentity	mmehost	mmerealm	UE-Reachability Indicates whether the MME supports UE Reachability...
1	hss.5GLaboratory	5GLaboratory	0
2	epc.5GLaboratory	5GLaboratory	0

No hacía falta modificar el imsi en la tabla pdn ni la tabla users, ya que en este caso, el equipo utilizará todos los valores de identificación por defecto.

En OAI EPC + OAI eNB + UE comercial

Como es un escenario real en el que habrá que pasar por una seria de autenticación con el UE posteriormente, es más restrictivo el uso de las bases de datos que el caso anterior. La configuración de aquella se encuentra en lo siguiente:

- *Tabla "users"*

Es la tabla que se encarga de almacenar la información de los clientes. Para ello, ha sido cambiado el tipo de "key", "rand" y "OPc" de varbinary a binary. Además, se han insertado a esta tabla, los siguientes datos concordando con los de la tarjeta USIM que se vaya a ser usada en el dispositivo UE:

```
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 ('901700000023532', '33611123456', '35609204079299', NULL, 'NOT_PURGED', '120', '40000000', '100000000', '47', '0000000000', '100', 773E70DF5E2E9FF399E675FFE4EDBA4F, '1', '0', '', 0x00000000000000000000000000000000, '');
```

En donde imsi, junto a key, que es la clave, son datos proporcionados por el fabricante de USIM. Y el valor de OPc se dejaría en blanco, ya que será calculado automáticamente a partir de la siguiente fórmula:

$$OPC = AES_{128}(k_i, OP) \oplus OP \quad \text{(Ecuación 2)}$$

Para aplicar dicha fórmula, habrá que introducir en el archivo hss.conf el valor de OP, que también es un dato dado por el fabricante de USIM. A la hora de la autenticación entre el epc y el UE, se comprobará que se coincida el valor de OPC calculado de cada uno.

El valor de mmeidentity es puesto a 100, ya que tendría que ser el mismo que es usado en la tabla de mmeidentity.

- *Table "mmeidentity"*

Contiene informaciones del nombre de equipo y de los dominios correspondientes a MME. Han sido modificados en esta tabla, en donde idmmeidentity=100, el valor de mmehost a "epc.5GLaboratory" y el de mmerealm a "5GLaboratory".

- *Table "pdn"*

Guarda las informaciones que se asocian un subscriptor (IMSI) y un APN. Se ha añadido una fila nueva con el dato correspondiente a users_imsi introducido anteriormente en la tabla de users.

- *Table "pgw"*

Es el responsable del almacenamiento de las informaciones que están relacionadas con P-GW. Se ha cambiado la dirección ipv4 de la fila id=3(correspondiente al valor de pgw_id de la tabla pdn) a la dirección de ip de MME de este equipo: 192.168.11.1, mientras que la dirección ipv6 es puesto a 0, ya que no va a ser usado en este caso.

3.4.2 Configuración de HSS

El siguiente paso consiste en la configuración de HSS, para ello, se han modificado los archivos de configuración correspondientes antes de poder compilar HSS. Uno de los archivos que se ha cambiado es el archivo hss.conf, que se encuentra en el directorio"/usr/local/etc/oai". A continuación, se explicarán las distintas modificaciones que se han hecho en dicho archivo en el caso de OAI EPC + OAISIM y en el caso de escenario real (OAI EPC + OAI eNB + UE) mientras que lo demás es el mismo para ambos casos:

En OAI EPC + OAISIM

- hss.conf:

```
{ ## MySQL mandatory options

MYSQL_server ="127.0.0.1";

MYSQL_user = "root";

MYSQL_pass = "linux";

MYSQL_db = "oai_db";
```

Se han tenido en cuenta solamente el nombre de usuario y la contraseña configurada para el software MYSQL al hacer la instalación de hss previamente. Todos los demás parámetros se han dejado por defecto.

En OAI EPC + OAI eNB + UE comercial

- hss.conf:

```
{ ## MySQL mandatory options

MYSQL_server ="127.0.0.1";

MYSQL_user = "root";

MYSQL_pass = "linux";

MYSQL_db = "oai_db";

## HSS options

OPERATOR_key = "BEA24B2C69D90A0392B3226DACF038F8";

RANDOM = "true";

## Freediameter options

FD_conf = "/usr/local/etc/oai/freeDiameter/hss_fd.conf";

};
```

En este archivo, el valor de operator_key es un dato que es proporcionado por el fabricante de USIM, en este caso, es “BEA24B2C69D90A0392B3226DACF038F8”. Es importante que sea el mismo valor de OPC que se usa en la programación de las tarjetas, ya que se usará este operator_key y la clave key metida anteriormente en la base de datos para calcular OPc.

El siguiente archivo a modificar es hss_fd.conf. Guarda en él las rutas de los certificados de autenticación además de los puertos de escucha. Tanto en el escenario virtual como el real, ha habido que cambiar el nombre de la identificación del equipo y su dominio. Además el

parámetro “ConnectPeer” se ha tenido que estar comentado, ya que en esta versión de repositorio se ha solucionado la localización por nombres:

- hss_fd.conf:

```
Identity = "hss.5GLaboratory";

Realm = "5GLaboratory";



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

Por el último, se han creado los certificados para HSS. Se ha escrito el siguiente comando en una terminal nuevo estando en el directorio “/openair-cn/scripts/”:

```
check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter/ hss.5GLaboratory
```

Una vez terminado esto, se ha compilado HSS con los siguientes comandos:

```
./build_hss -c

./run_hss
```

3.4.3 Configuración de MME

En este apartado, se ha procedido a modificar los archivos de la configuración de mme para poder lanzarlo posteriormente. En el mismo directorio que hss.conf, se encuentra el archivo mme.conf, al igual que el apartado anterior, se van a hablar de los parámetros que han sido modificados de este archivo en el escenario virtual y el del laboratorio:

En OAI EPC + OAISIM

- mme.conf:

```
   REALM = "5GLaboratory" ;

  NETWORK_INTERFACES :

MME_INTERFACE_NAME_FOR_S1_MME            = "ens33”          MME_IPV4_ADDRESS_FOR_S1_MME
= "192.168.1.1/24”        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 OAI EPC + OAI eNB + UE comercial

- mme.conf:

```
 REALM = "5GLaboratory";

   GUMMEI_LIST = (

{MCC="901" ; MNC="70"; MME_GID="4" ; MME_CODE="1";

   TAI_LIST = (

{MCC="901" ; MNC="70"; TAC = "1";

 );

 NETWORK_INTERFACES :

MME INTERFACE NAME FOR S1 MME            = "eno1"          MME_IPV4_ADDRESS_FOR_S1_MME
= "192.168.11.1/24"        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 donde MCC y MNC son los datos que concuerdan con los de la tarjeta USIM que se vaya a usar.

A continuación se ha modificado el archivo mme_fd.conf que contiene las informaciones del nombre y del dominio del equipo, además indica el puerto al que está conectado mme y las rutas de los certificados generados para la autenticación de mme.

- mme_fd.conf:

```
Identity = "epc.5GLaboratory";

Realm = "5GLaboratory";

Port = 3870;

SecPort = 5870;


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

Una vez acabada la configuración de mme, se ha generado el certificado de la autenticación de la misma manera que en hss: ejecutando el siguiente comando en el directorio “/openair-cn/scripts/”:

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

En una terminal nueva, se ha compilado mme:

```
./build_mme -c

./run_mme
```

3.4.4 Configuración de spgw

La última modificación de la configuración del componente de EPC que quedaba era la de S-PG. Se ha modificado el archivo spgw.conf. Los parámetros modificados en dicho archivo de cada escenario se muestran en lo siguiente:

En OAI EPC + OAISIM

- spgw.conf:

```
   NETWORK_INTERFACES :

    {

        SGW INTERFACE NAME FOR S11              = "lo

        SGW IPV4 ADDRESS FOR S11                = "127.0.11.2

        SGW INTERFACE NAME FOR S1U S12 S4 UP    = "ens33";

        SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP      = "192.168.1.1/24

        SGW IPV4 PORT FOR S1U S12 S4 UP         = 2152

};
```

En OAI EPC + OAI eNB + UE comercial

- spgw.conf:

```
   NETWORK_INTERFACES :

    {

        SGW INTERFACE NAME FOR S11              = "lo

        SGW IPV4 ADDRESS FOR S11                = "127.0.11.2

        SGW INTERFACE NAME FOR S1U S12 S4 UP    = "eno1";

        SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP      = "192.168.11.1/24

        SGW IPV4 PORT FOR S1U S12 S4 UP         = 2152
```

```
};

P-GW =

NETWORK_INTERFACES :

{        PGW_INTERFACE_NAME_FOR_SGI         = "wlp2s0";

        PGW_MASQUERADE_SGI                 = "yes"

        UE_TCP_MSS_CLAMPING                = "no";                              };
```

A diferencia del escenario virtual, en este caso, para un correcto acceso a internet una vez que el UE se conecte al eNB, es necesario que esté activada la opción de la mascarada de P-GW, además la interfaz que se utiliza para el encaminamiento de PGW es wlp2s0, en nuestro caso, la red wifi.

No hacía falta generar certificados de autenticación a diferencia de otros dos. Y se ha compilado spgw en otra terminal nueva para terminar la parte de la configuración de EPC:

```
./build_spgw –c

./run_spgw
```

3.5 Instalación de OAISIM

Al igual que en EPC, se ha descargado el repositorio correspondiente desde eurecom.fr para la máquina virtual llamada OAISIM. Dicho repositorio incluirá además de los scripts de OAISIM, también los comandos de ejecución para el eNB y el UE. Por lo tanto, es el mismo repositorio que se utiliza tanto para el OAISIM como para el OAI eNB.

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

Estando en el directorio “/openairinterface5g/cmake_targets/”, se han instalado unos paquetes con el siguiente comando:

```
./build_oai -I
```

Solo es necesario realizar una única vez la instalación de éste.

3.5.1 Configuración de OAISIM

Una vez terminada la instalación del repositorio, se ha seguido con la modificación del archivo de configuración que se encuentra en el directorio "/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.generic.oaisim.local mme.conf". A continuación, se plasman los parámetros modificados de dicho fichero:

- enb.band7.generic.oaisim.local mme.conf

```
////////// MME parameters:

mme_ip_address      = ( {

ipv4       = "192.168.1.1";

preference = "ipv4";

})

NETWORK_INTERFACES :

{

ENB_INTERFACE_NAME_FOR_S1_MME     = "ens34";

ENB IPV4 ADDRESS FOR S1 MME       = "192.168.1.2/24";

ENB INTERFACE NAME FOR S1U        = "ens34";

ENB_IPV4_ADDRESS_FOR_S1U          = "192.168.1.2/24";

ENB PORT FOR S1U                  = 2152;

};
```

Tras esto, estando en el directorio "/openairinterface5g/cmake_targets", se ha compilado el siguiente comando:

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

En donde:

--oaisim: crea un simulador de OAISIM. El hardware está definido como ninguno por defecto

--UE: crea las partes específicas de UE

-x: agrega una función de osciloscopio de software a los binarios producidos

-c: elimina todos los ficheros compilados anteriormente

Por el último, se ha procedido lanzar la aplicación. En este modelo que se ha usado, se ha recreado un UE y un eNB, ambos virtualizados. El OAISIM está conectado al EPC mediante la interfaz S1. Se ha ejecutado el siguiente comando dentro del directorio "/openairinterface5g/cmake_targets/tools":

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

3.5.2 Instalación de OAI eNB

Se ha repetido el proceso de la instalación del repositorio “openairinterface5g” que el apartado…

3.5.3 Instalación de la librería de bladeRF

Siendo un escenario real, se ha instalado la librería de bladeRF para poder usarlo como estación base, también se ha tenido en cuenta que el dispositivo bladeRF debería estar conectado a un puerto USB 3.0 para el correcto funcionamiento.

```
sudo add-apt-repository ppa:bladerf/bladerf

sudo apt-get update

sudo apt-get install bladerf

sudo apt-get install libbladerf-dev

sudo apt-get install bladerf-firmware-fx3

sudo apt-get install bladerf-fpga-hostedx40   # for the 40 kLE hardware
```

Una vez finalizada la instalación, se ha procedido a hacer la compilación de bladeRF como eNB, para ello, se ha situado en el directorio “/openairinterface5g/cmake_targets/”:

```
./build_oai -c -x -w BLADERF --eNB
```

En donde:

--eNB: crea el simulador lte-softmodem de eNB.

-x: agrega una función de osciloscopio de software a los binarios producidos

-w: añade el soporte de hardware, en este caso es bladeRF

-c: elimina todos los ficheros compilados anteriormente

3.5.4 Configuración de OAI eNB

El archivo que se encarga de la configuración de bladeRF como el eNB se encuentra en el directorio“/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.bladeRF.conf”. Los parámetros modificados para estar de acuerdo con las configuraciones anteriores del EPC son los siguientes:

- enb.band7.tm1.bladeRF.conf:

```
tracking_area_code =  "1";

  mobile_country_code = "901";

  mobile network code= "70";

  ipv4       = "192.168.11.1";


  NETWORK_INTERFACES :

  {

      ENB INTERFACE NAME FOR S1 MME     = "enp0s31f6";

      ENB IPV4 ADDRESS FOR S1 MME            = "192.168.11.2/24";

      ENB INTERFACE NAME FOR S1U               = "enp0s31f6";

      ENB IPV4 ADDRESS FOR S1U                      = "192.168.11.2/24";

      ENB PORT FOR S1U                                      = 2152;

};
```

Tras la modificación de dicho archivo, se ha lanzado la aplicación con el siguiente comando estando en una terminal nueva:

```
./cmake_targets/lte_build_oai/build/lte-softmodem -d -O ./targets/PROJECTS/GENERIC-
LTE-EPC/CONF/enb.band7.tm1.bladerfx40.conf
```

3.6 Programación de la tarjeta

Cuando estén los dos equipos en marcha, se ha hecho la programación de las tarjetas USIM de Sysmocom como se van a explicar en los siguientes pasos.

El programa que se ha utilizado en este proyecto se llama pySim-prog, consiste en una pequeña utilidad de línea de comandos que está escrita en python. Como se había explicado en el apartado 2, las tarjetas USIM son tarjetas especiales que a diferencia de las tarjetas comerciales, utilizan un tipo de claves que les permiten escribir los archivos/campos que normalmente solo un operador puede programar. [2]

Los datos de la tarjeta USIM que se ha usado, se plasman en la tabla 3.5.

IMSI	901700000023532
ICCID	8988211000000235328
ACC	0004
Ki	773E70DF5E2E9FF399E675FFE4EDBA4F
OPC	BEA24B2C69D90A0392B3226DACF038F8

Tabla 3. 5 Parámetros USIM

Para empezar, se ha instalado la dependencia con el equipo Linux:

```
sudo apt-get install pcscd pcsd-tools libccid libpcsclite-dev python-pyscard
```

Tras esto, se ha conectado el lector de tarjetas al equipo con la tarjeta USIM insertada al ordenador mediante un cable USB.

Posteriormente, se han descargado los códigos fuentes del programa pySIM.

```
git clone git://git.osmocom.org/pysim pysim
```

Finalmente, se ha procedido a realizar la programación con el siguiente comando:

```
./pySim -prog.py -p 1 -t Magic -SJS1 -a 85601364 -x 901 -y 70 -i 901700000023532 -s
8988211000000235328 --op=BEA24B2C69D90A0392B3226DACF038F8 –k
773E70DF5E2E9FF399E675FFE4EDBA4F
```

Es importante que los datos introducidos al programa se concuerden con los proporcionados por el fabricante de la tarjeta USIM.

Una vez tenida la tarjeta USIM programada correctamente, se ha mostrado por consola la siguiente información:

```
Generated card parameters :
 > Name    : Magic
 > SMSP    : e1ffffffffffffffffffffffff0581005155f5ffffffffffff000000
 > ICCID   : 8988211000000235328
 > MCC/MNC : 901/70
 > IMSI    : 901700000023532
 > Ki      : 773E70DF5E2E9FF399E675FFE4EDBA4F
 > OPC     : 03a960a688825e4dfea74352f3489105
 > ACC     : None

Programming ...
Done !
```

Figura 3. 7 Resultado de la programación de USIM

3.7 Configuración del dispositivo UE

El otro dispositivo a configurar es el dispositivo móvil que servirá de UE, en este caso, se ha usado un HUAWEI P9. En primer lugar, se ha insertado la tarjeta Usim programada previamente al móvil. Tras esto, se ha creado un APN (nombre de punto de acceso) en la ventana de redes móviles como se muestra abajo:

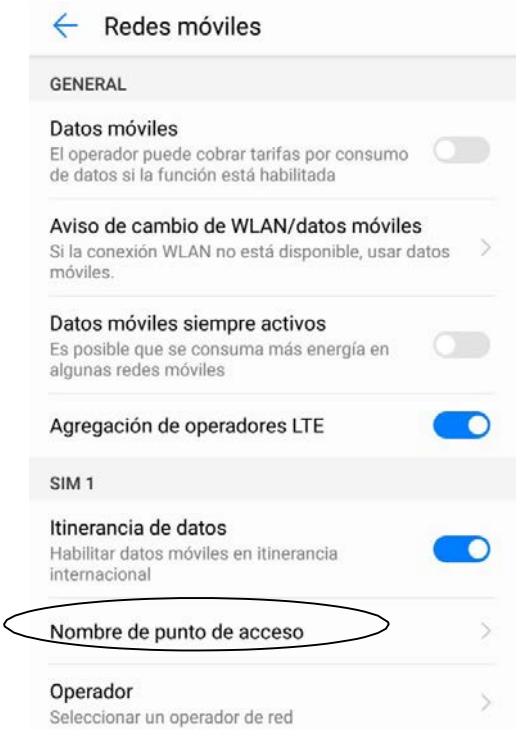

Figura 3. 8 Configuración de APN: paso 1

La configuración de este APN creado debería coincidirse con los datos configurados en el equipo EPC y eNB en los apartados anteriores. Los parámetros modificados se muestran en las siguientes figuras:

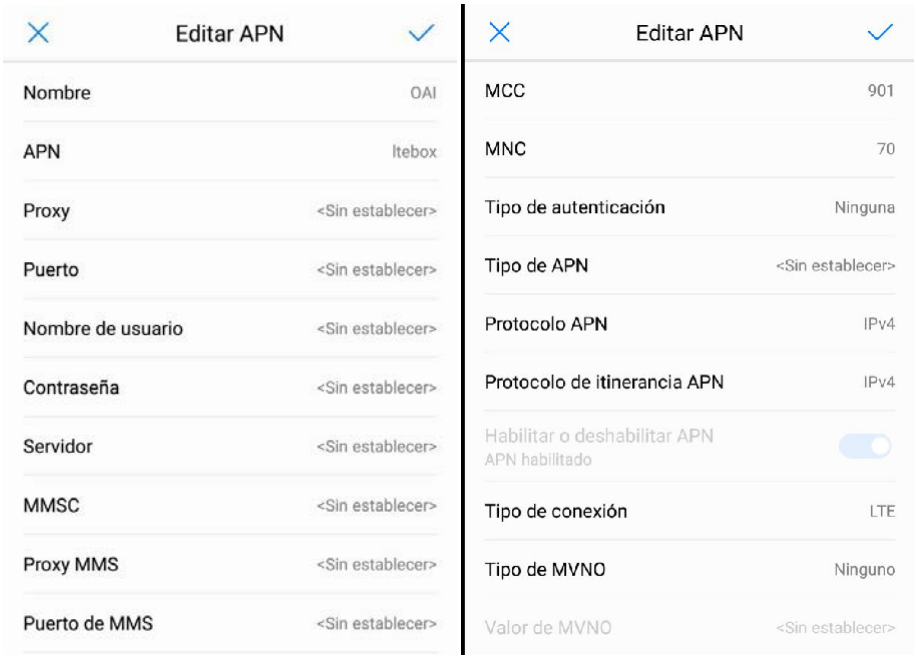

Figura 3. 9 Configuración de APN: paso 2

Finalmente, una vez tenido el APN creado, se ha aparecido en la lista el nuevo APN con el nombre dado por nosotros como se destaca en la figura siguiente:

Figura 3. 10 Resultado de la configuración de APN

3.8 En OAI EPC + OAI eNB + OAI UE

Una vez conseguido que OAI EPC + OAI eNB + UE comercial se hayan puesto en marcha, se ha procedido a sustituir el UE comercial, es decir, el dispositivo móvil, por un OAI UE usando otro ordenador con OpenAirInterface, y conectado éste a otro driver bladeRFx40 a través de USB 3.0. El OAI UE con la estación base eNB se comunicará a través de dos antenas.

Como se ha usado la misma base que OAI EPC + OAI eNB + UE comercial, se ha añadido solamente la instalación y la configuración de OAI UE. Se ha instalado Ubuntu 16.04.1 TSL en un ordenador nuevo, en este caso, se ha usado uno de la marca DELL de la serie Latitude E5550, asegurándonos de que éste cumpliera las especificaciones requeridas para OAI. Tras esto, se ha realizado la preparación de equipo siguiendo los pasos explicados en el apartado…

A continuación, se ha descargado el mismo repositorio “openairinterface5g” como para eNB:

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

Para continuar, se ha habilitado Linux y Netlink para poder hacer transferencias de datos entre UE y eNB. Para ello, se han modificado las siguientes dos líneas en “CMakeLists.txt”:

En lugar de:

add_boolean_option(LINUX False “used in weird memcp() in pdcp.c ???”)

se tiene:

add_boolean_option(LINUX True “used in weird memcp() in pdcp.c ???”)

En lugar de:

add_boolean_option(PDCP_USE_NETLINK False “For eNB, PDCP communicate with a NETLINK socket if connected to network driver, else could use a RT-FIFO ”)

se tiene:

add_boolean_option(PDCP_USE_NETLINK True "For eNB, PDCP communicate with a NETLINK socket if connected to network driver, else could use a RT-FIFO ")

Una vez finalizada la modificación, se ha procedido a descargar la librería para bladeRF como se había explicado anteriormente. Y se ha conectado la segunda tarjeta bladeRF a través de USB 3.0.

En segundo lugar, se han descargado los paquetes necesarios para OAI:

```
sudo ./openairinterface5g/cmake_targets/build_oai –I
```

Tras esto, se ha realizado la compilación de OAI utilizando a bladeRF como UE:

```
sudo ./openairinterface5g/cmake_targets/build_oai -w BLADERF --UE
```

Por el último, antes de ejecutar UE, se ha tenido que instalar el modulo de kernel: ue_ip.ko mediante init_nas_s1, además éste se encarga de la creación de la interfaz OIP, a la que la entidad MME se realiza la asignación de las direcciones IP.

```
cd ~/openairinterface5g/

sudo ./targets/bin/init_nas_s1 UE
```

Debido a la actualización de las librerías de "openairinterface5g", el comando anterior para la compilación del archivo init_nas_s1 ya no se utiliza, para ello, se han compilado los comandos contenidos en dicho archivo manualmente de la siguiente manera:

```
sudo ifconfig oip0 up

sudo insmod ue_ip.ko

sudo ip route flush cache

sleep 1

sudo sysctl -w net.ipv4.conf.all.log_martians=1

sudo echo "Disabling reverse path filtering"

sudo sysctl -w net.ipv4.conf.all.rp_filter=0

sudo ip route flush cache

sudo fgrep lte /etc/iproute2/rt_tables > /dev/null

if [ $? -ne 0 ]; then

echo "200 lte " >> /etc/iproute2/rt_tables

sudo ip rule add fwmark 1 table lte

sudo ip route add default dev $LTEIF table lte
```

Además, para sustituir la tarjeta USIM que se había utilizado en un UE comercial, se han tenido que realizar las siguientes modificaciones para que los datos de este UE concuerden con la configuración de la base de datos:

En el fichero "openairinterface5g/openair2/UTILS/mcc_mnc_itu.c" se ha añadido nuestro MCC y MNC:

```
{901, "70"};
```

Por el otro lado, se han añadido en los siguientes ficheros datos que concuerdan con la tarjeta USIM, es decir, el IMSI, la clave de seguridad key, el código de operator OPC, y etc.

- /openairinterface5g/cmake_targets/build_oai
- /openairinterface5g/openair3/NAS/TOOLS/network.h
- /openairinterface5g/openair3/NAS/TOOLS/ue_eurecom_test_sfr.conf
- /openairinterface5g/openair3/NAS/TOOLS/ue_tcl_test.conf"

Se ha ejecutado primero el equipo EPC como se habían explicado en el apartado … y luego para lanzar la aplicación de eNB, se ha variado ligeramente el comando. Dentro del directorio "openairinterface5g" se ha ejecutado lo siguiente:

```
sudo -E ./targets/bin/lte-softmodem.Rel14 –O ./targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.bladeRFx40
```

Una vez teniendo el equipo de EPC y el equipo de eNB en marcha, se ha lanzado el comando que se muestra a continuación para la compilación de UE:

```
cd openairinterface5g/targets/bin

sudo -E ./lte-softmodem.Rel14 -U -C2680000000 -r25 --ue-txgain 60 --ue-rxgain 120
```

Capítulo 4: Análisis de resultados

En este capítulo, se estudiarán individualmente los resultados de la implementación de los tres escenarios realizados en el capítulo…. Para ello, se han empleado las herramientas como wireshark, el emulador del osciloscopio: Soft-Scope, el analizador de tramas: T-tracer y un software llamado Fast Speed Test en el equipo de usuario para analizar el test de velocidad de internet.

4.1 OAI EPC + OAISIM

Wireshark

Este es un escenario virtualizado, en el que no se ha utilizado ningún dispositivo real, es decir, se ha encargado el simulador OAISIM de las funciones que se realizan el eNB y el UE. Usamos la herramienta wireshark para ver cómo se transmiten los datos, para ello, se ha tenido en cuenta que la dirección IP asignada al equipo de OAISIM es 192.168.1.2 mientras que en MME es 192.168.1.1 como lo que había explicado en el capítulo 3:

No.	Time	Source	Destination	Protocol	Length	Info
69	50.530010124	192.168.1.2	192.168.1.1	S1AP	124	id-S1Setup, S1SetupRequest
71	50.531209887	192.168.1.1	192.168.1.2	S1AP	92	id-S1Setup, S1SetupResponse
74	55.341493271	192.168.1.2	192.168.1.1	S1AP/N…	160	id-initialUEMessage, Attach request, PDN connectivit…
85	55.355474652	192.168.1.1	192.168.1.2	S1AP/N…	144	SACK id-downlinkNASTransport, Authentication request
87	55.490113708	192.168.1.2	192.168.1.1	S1AP/N…	140	SACK id-uplinkNASTransport, Authentication response
88	55.491359019	192.168.1.1	192.168.1.2	S1AP/N…	120	SACK id-downlinkNASTransport, Security mode command
89	55.599042392	192.168.1.2	192.168.1.1	S1AP/N…	136	SACK id-uplinkNASTransport, Security mode complete
106	55.626528668	192.168.1.1	192.168.1.2	S1AP/N…	276	SACK id-InitialContextSetup, InitialContextSetupRequ…
109	55.856131232	192.168.1.2	192.168.1.1	S1AP	112	id-UECapabilityInfoIndicationUECapabilityInformation
111	56.065736068	192.168.1.2	192.168.1.1	S1AP	104	id-InitialContextSetup, InitialContextSetupResponse

Figura 4. 1 Intercambio de señalización en escenario virtual

Para empezar, comienza el eNB virtualizado mandando S1SetupRequest a MME, que consiste en una petición del establecimiento. Una vez que es recibido por MME, lo confirma para establecer la conexión entre los dos equipos.

Tras esto, el UE envía el *Attach request*, en el cual incluye toda la información sobre la identidad del UE, al MME. Una vez recibido el MME el anterior mensaje, comienza todo el proceso de la autenticación y seguridad. Para ello, se tiene que comprobar que la clave de seguridad proporcionada por el UE coincide con la que es generada mediante el algoritmo EPS AKA. En primer lugar, MME le envía al eNB el *Authentication request*, y éste se lo reenviará en seguida al UE. UE preparará la respuesta del desafío con el parámetro RES en el mensaje *Authentication Response y se lo mandará a MME. Tras esto, se completa el proceso de autenticación.* Con el fin de ordenar al UE que active la seguridad AS, MME le manda *Security mode command* al UE que acabará recibiendo por parte de UE, el *Security mode complete*. [17]

4.2 OAI EPC + OAI eNB + UE (comercial)

Wireshark

Figura 4. 2 Intercambio de señalización en el escenario laboratorio

La figura 4.2 muestra el intercambio de mensajes entre la estación base, en este caso, que es la que tiene la dirección IP 192.168.11.2 y el equipo que simula de la red troncal EPC, cuya dirección IP asignada es 192.168.11.1. Los mensajes que se intercambian siguen los mismos procesos explicados en el escenario virtualizado.

Monitorización

Una vez que se establezca la conexión entre UE y MME a través de la tarjeta de desarrollo BladeRFx40, por pantalla de los equipos se pueden ver lo siguiente:

Figura 4. 3 Monitorización de eNB tras la correcta conexión con EPC

Figura 4. 4 Monitorización de MME tras la correcta conexión con eNB

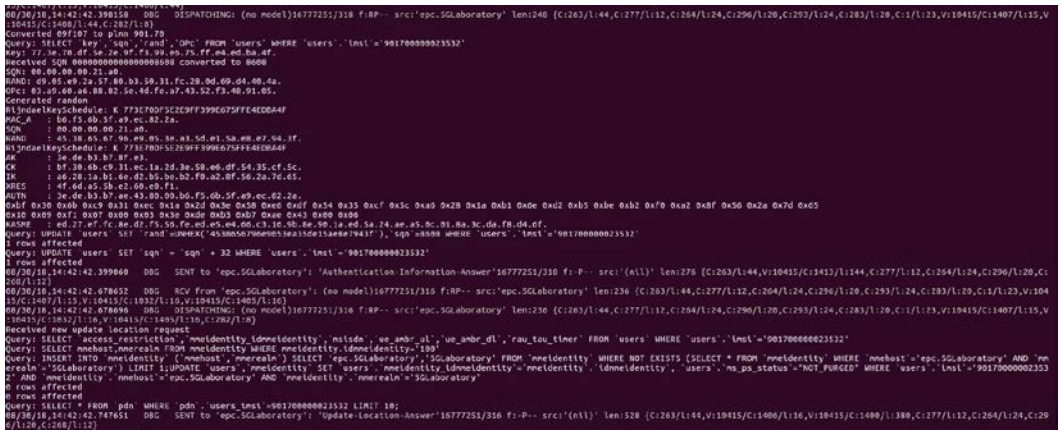

Figura 4. 5 Monitorización de HSS tras la correcta conexión con eNB

La figura 4.3 pertenece al log de eNB una vez que se conecte con el UE, se puede ver el identificador de dicho UE que está conectado a la estación base. El estado de MME como se muestra en la figura 4.4, se puede observar que está conectado al MME tanto el eNB como el UE. Y por el último, la figura 4.5 indica cuando MME detecta la presencia del UE, y en seguida, el equipo HSS comienza el proceso de la autenticación.

La figura 4.6 muestra el Scope de la estación base eNB para el Uplink. Los datos que muestra Scope, son las siguientes:

- Señal recibida en el dominio del tiempo (dB)
- Respuesta frecuencial del canal (dB)
- Pusch (Physical Uplink Shared Channel) log-likelihood
- Constelación del Pusch
- Throughput del Pusch

Se observa que la constelación es 16 QAM.

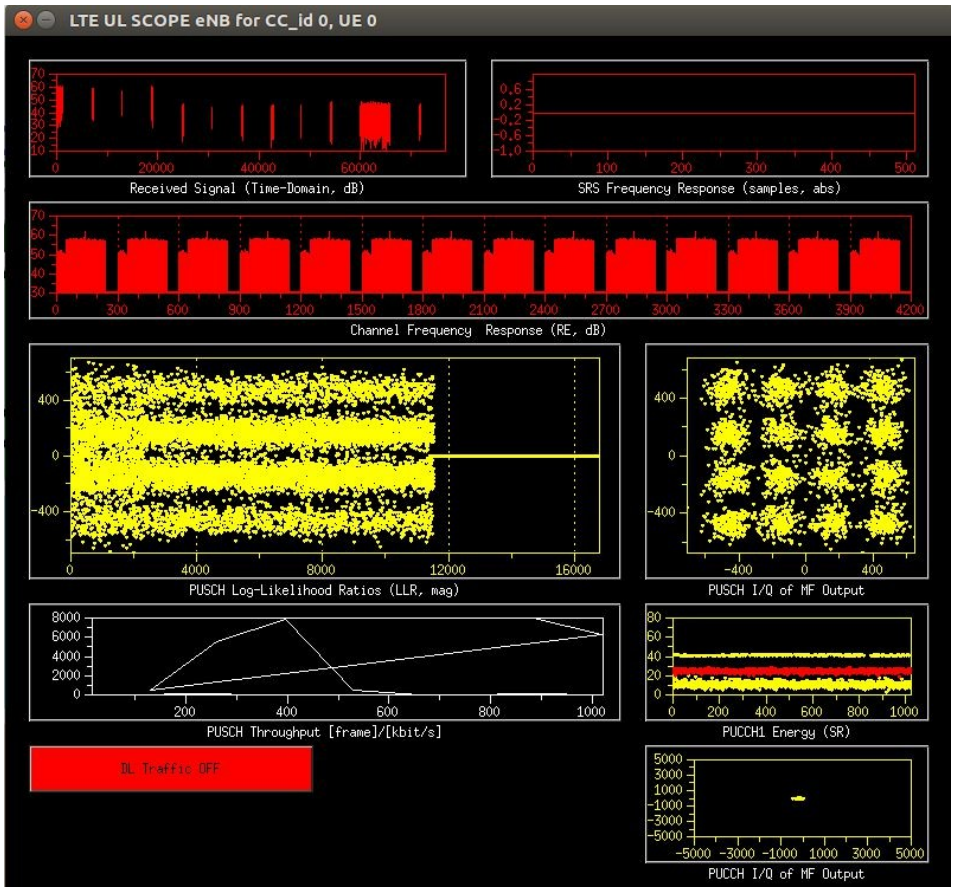

Figura 4. 6 Herramienta OAI soft-scope

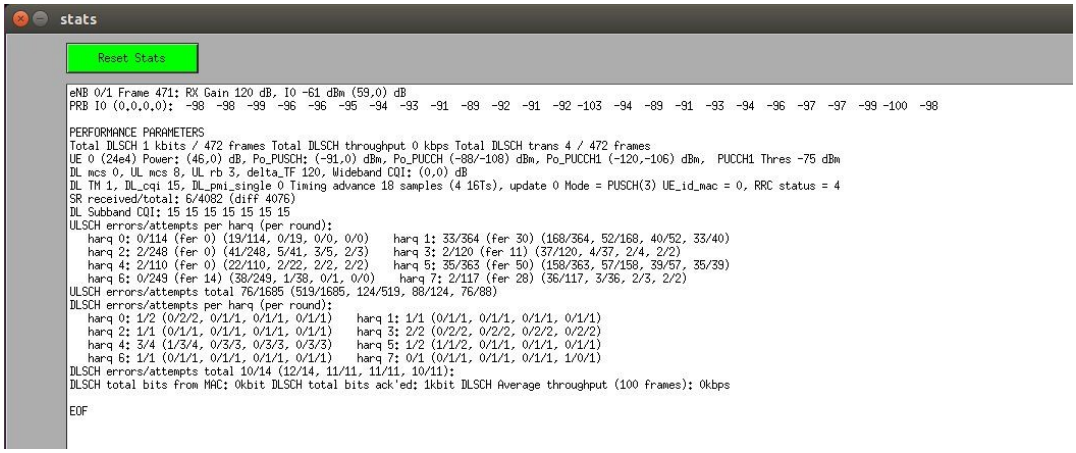

Figura 4. 7 Estadísticas de OAI Soft-Scope

Las estadísticas principales son estas: [12]

- DLSCH (Downlink Shared Channel):
  - Tasa de bits transmitidos correctamente
  - Throughput
- Usuario:
  - Potencia Pusch

- ☐ Potencia PUCCH (Physical Uplink Control Channel)
- MCS (Modulation and Coding Scheme)
- Modo de transmisión
- Procesos HARQ

También se ha utilizado la herramienta T-tracer en la parte de eNB para capturar las tramas que se envían en las distintas capas de UE como se muestra en la figura 4.8

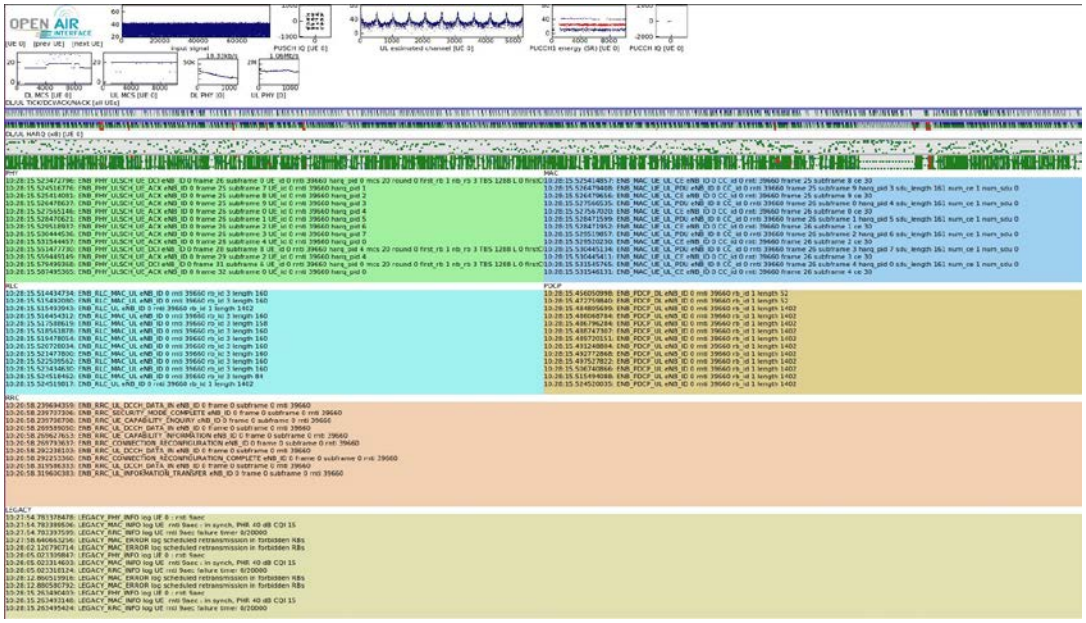

Figura 4. 8 Analizador de tramas T-tracer

Lo que se presenta en la figura es el momento en el cual UE estaba registrado en la red creada del eNB, y consiguió a navegar correctamente por el internet. Se puede ver que en este momento, se han enviado una gran cantidad de datos del tipo ACK/NACK, TICK y HARQ tanto en el uplink como en el downlink del UE. En el momento que se hizo la captura, la velocidad de bajada de eNB es 18.33 Kb/s mientras que la velocidad de subida del mismo es 1.06 Mb/s. Se observa además una constelación 16-QAM, lo mismo que lo que se había visto en el Soft-Scope.

Test de velocidad

Por el último, se ha hecho un test de velocidad de internet usando el software *Fast Speed Test* una vez que el equipo de usuario UE esté registrado en la red y que tenga acceso a internet gracias a la entidad P-SW del equipo EPC. En la figura 4.9 Figura 4. 9 Primera prueba del test de velocidady la figura 4.10 Figura 4. 10 Segunda prueba del test de velocidadse muestran las dos pruebas que se han hecho. Como se ven, en el servidor de red que está en la barra de arriba, pone 901704G, que es la red LTE creada en nuestro caso, el nombre está compuesto por el código MCC y el MNC. Lo que se ha hecho este software es hacer un ping desde la dirección IP asignada por nuestra red al equipo de usuario UE a la dirección IP *217.126.198.179* que pertenece a la empresa *Telefónica* localizada en Madrid.

Figura 4. 9 Primera prueba del test de velocidad

Figura 4. 10 Segunda prueba del test de velocidad

n ambas pruebas, la velocidad de descarga es buena mientras que la de subida es mucho peor.

En la figura 4.11 se muestra una comparación del test de velocidad de internet entre la red LTE creada y la red WIFI que se ha usado el equipo EPC.

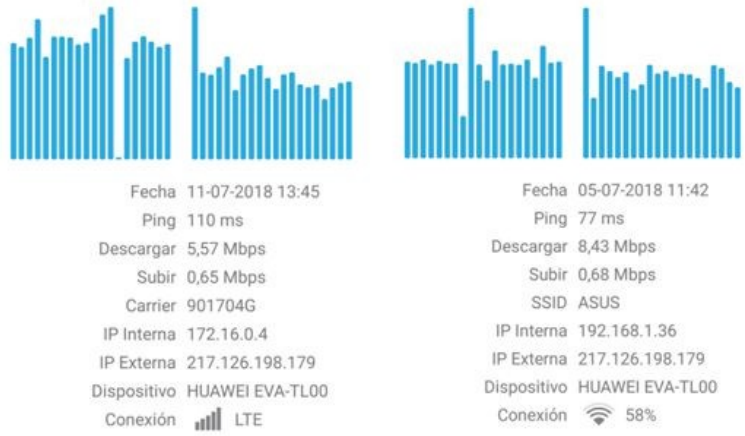

Figura 4. 11 Comparación de velocidad de internet entre red celular LTE creada y wifi

Como se había comentado anteriormente, la velocidad de descarga de LTE es mucho mejor que la de subida, que es una característica típica en las redes de 4G. Comparando las velocidades de las dos redes, se puede ver que la red LTE creada ha alcanzado un 66.08% de la velocidad de descarga de wifi en este caso, lo cual es aceptablemente buena.

4.3 OAI EPC + OAI eNB + OAI UE

Desafortunadamente, no ha habido éxito con la implementación de este escenario. En primer lugar, después de lanzar la ejecución de OAI UE, el UE le envía un RRCCONNECTION REQUEST a eNB y espera a que el eNB le conteste con el msg4 que es una copia de los últimos 6 bits de la trama enviada por el UE anteriormente, parece ser que eNB llega a enviárselo, pero UE espera cierto tiempo, como no lo recibe y no puede mandar la confirmación a eNB, por eso, el eNB al final lanza una liberación a UE. En el anexo A y B se plasman los logs de ambos dispositivos en este caso. Una posibilidad de esta falta de sincronización entre eNB y UE podría ser que el eNB funciona a potencia fija, mientras que el UE va regulando la potencia durante la conexión. Como la tarjeta de desarrollo bladeRF no tienen las ganancias bien programadas, no se sincroniza correctamente el UE con el eNB.

En segundo lugar, se ha intentado mirar las funciones que se encargan de la programación de las ganancias de transmisor y receptor dentro de las librerías de bladeRF, pero debido a la inestabilidad de la nueva librería de la tarjeta con el software *Openairinterface5g*, tampoco ha habido éxito.

Capítulo 5: Presupuestos

En este capítulo, se ha hecho un cálculo aproximado de los costes involucrados en este proyecto según los equipos que se han utilizado, que se detallan en la tabla 5.1 y los costes de los recursos humanos, que se mostrarán en la tabla 5.2.

Recursos hardware	Coste/unidad(€)	Unidad	Coste total(€)
Pórtatil Dell Latitude E5570	1190	1	1.190
Pórtatil Dell Latitude E5540	1020	1	1.020
Pórtatil Dell Latitude E5550	994	1	994
Discos duros externos	48.89	3	146,67
bladeRFx40	420	2	840
Antenas	100	4	400
Tarjetas USIM de Sysmocom	4.76	10	47,6
Equipo de usuario(Huawei P9)	353	1	353

Tabla 5. 1 Estimación de costes de recursos hardware

Para la implementación de escenarios, se ha utilizado el sistema operativo Linux que está instalado en los discos duros externos. Se han utilizado el máximo número de recursos hardware para el escenario de laboratorio: OAI EPC + OAI eNB + OAI UE: tres ordenadores, tres discos duros, dos tarjetas de desarrollo bladeRFx40 y cuatro antenas. En cuanto al equipo de usuario comercial, se ha utilizado un móvil "Huawei P9" que soporta el funcionamiento de LTE y trabaja en la banda requerida de 2.68 GHz, lo mismo pasan con las cuatro antenas.

Recursos Humanos	Coste/h (€)	Tiempo dedicado (h)	Coste total (€)
Xueyan Xiang	3.43	400	1.376
Zaloa Fernández	8	20	160

Tabla 5. 2 Estimación de costes de recursos humanos

Finalmente, se ha calculado la estimación del coste total teniendo en cuenta los dos recursos anteriores:

Coste total: Recuros hardware (€)	Coste total: Recuros humanos (€)	Coste final (€)
4991.27	1536	6527.27

Tabla 5. 3 Estimación final de costes

Capítulo 6: Conclusiones

En este proyecto se ha podido estudiar con profundidad los conocimientos de la tecnología LTE, y también se han implementado tres tipos de escenario usando el software libre OpenAirInterface.

El primero consiste en un escenario virtualizado, que se realizó con éxito, y se ha podido analizar el intercambio de la señalización entre la máquina virtual de la red troncal EPC y la máquina virtual de OAISIM, en la cual se virtualiza tanto la estación base eNB como el equipo de usuario UE.

Tras esto, se trasladó la idea de la implementación del escenario virtualizado en uno de laboratorio, utilizando una tarjeta de desarrollo bladeRF para que funcione como la estación base de la red celular LTE, y un dispositivo móvil para el equipo de usuario. También se han obtenido buenos resultados en este caso, en el que se pudo hacer un test de velocidad de acceso a servicio a internet. Con la máxima velocidad de descarga que se obtuvo, fue incluso posble ver videos sin ningún problema.

Por el último, en cuanto al tercer escenario que se ha implementado, se trata de sustituir el dispositivo móvil comercial por un equipo de usuario virtualizado utilizando OAI. Debido a la mala sincronización entre OAI eNB y OAI UE, el resultado no ha sido bueno.

De todas las pruebas que se han hecho, se ha podido ver que la plataforma OAI, siendo una plataforma abierta y de código libre, en general es una herramienta muy útil para investigación de las tecnologías de 4G y 5G a medio-largo plazo aunque también hay que admitir que tiene otras debilidades, las cuales le limitan ser usado para el despliegue de una red celular LTE a bajo coste en la vida real. Por ejemplo, una de las desventajas que tiene la plataforma OAI, es que debido a las altas ambiciones de las personas detrás de ella, lo hacen compleja, y dado que está en rápido desarrollo, podría también demuestra ser inestables. Además, actualmente no es compatible con más de unos pocos UEs a la vez.

Acrónimos:

3GPP	Third Generation Partnership Project
AM	Acknowledge Mode
AN	Access Network
APN	Access Point Name
APN-AMBR	APN Aggregated Maximum Bit Rate
ARP	Allocation and Retention Priority
BCCH	Broadcast Control Channel
CCCH	Common Control Channel
CN	Core Network
CQI	Channel Quality Indicator
DCCH	Dedicated Control Channel
DCI	Downlink Control Indicator
DLSCH	Downlink Shared Channel
DNS	Domain Name System
DTCH	Dedicated Traffic Channel
eNB	Evolved Node-B
EMM EPS	Mobility Management
EPC	Evolved Packet Core
EPS	Evolved Packet System
ESM	EPS Session Management
ETX	Expected Transmission Count
E-UTRAN	Evolved Terrestrial Radio Access Network
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
GBR	Guaranteed Bit Rate
GSM	Global System for Mobile

GRCM	Grup de Recerca en Comunicacions Mobile
GUTI	Globally Unique Temporary Identifier
HARQ	Hybrid Automatic Repeat Request
HSPA	High-Speed Packet Access
HSS	Home Subscriber Server
ICIC	Inter-Cell Interference Coordination
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol LTE Long Term Evolution
MAC	Medium Access Layer
MaxC/I	Maximum-Carrier-over-Interference
MBR	Maximum Bit Rate
MCC	Mobile Country COde
MCS	Modulation and Coding Scheme
MME	Mobility Management Entity
MNC	Mobile Network Code
MTU	Maximum Transmission Unit NAS Non-access stratum
NAT	Network Adress Translation
OAI	OpenAirInterface
OFDMA	Orthogonal Frequency Division Multiple Access
OSA	OpenAirInterface Software Alliance
PBCH	Physical Broadcast Channel
PCC	Policy and Charging Control
PCFICH	Physical Control Format Indicator Channel
PCRF	Policy and Charging Rule Function
PD	Proportional Demand
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol

PDN	Packet Data Network
PDSCH	Physical Downlink Shared Channel
PDU	Protocol Data Unit PF Proportional Fair
PGW	Packet Data Network Gateway
PHICH	Physical Hybrid-ARQ Indicator Channel
PHR	Power Headroom
PHY	Physical Layer
PMCH	Physical Multicast Channel
PRACH	Physical Random Access Channel
PS	Packet switching
PSS	Primary Synchronization Signal
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
QCI	QoS Class Identifier
Qos	Quality of Service
RAN	Radio Acces Network
RB	Radio Bearer
RLC	Radio Link Control
RNTI	Radio Network Temporary Identifier
RR	Round Robin RRC Radio Resource Control
RRM	Radio Resource Management
RTT	Round Trip Time
SC-FDMA	Single Carrier – Frequency Division Multiple Access
SDR	Software Defined Radio
SDU	Service Data Unit
SGI	Service Gateway Interface
SGW	Serving Gateway

SIB	System Information Block
SIM	Subscriber Identity Module
SPGW	Serving Packet Data Network Gateway
SRB	Signalling Radio Bearer
SSS	Secondary Synchronization Signal
TAC	Tracking Area Code
TB	Transport Block
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TFT	Traffic Flow Template
TM	Transparent Mode
TSC	Teoria de Senyal i Comunicació
TTI	Transmission Time Interval
UE	User Equipment
UE-AMBR	UE Aggregated Maximum Bit Rate
UHD	USRP Hardware Driver
UM	Unacknowledged Mode
UMTS	Universal Mobile Telecommunications System
USIM	Universal SIM
USRP	Universal Software Radio Peripheral

Referencias:

[1] BladeRF. 2018 GitHub, Inc.

https://github.com/Nuand/bladeRF/wiki/Getting-Started%3A-Verifying-Basic-Device-Operation

[2] pySim Wiki.

https://osmocom.org/projects/pysim/wiki

[3] Patch for EPC and eNB. 2017.

https://open-cells.com/index.php/2017/08/22/all-in-one-openairinterface-august-22nd/

[4] Página oficial de OAI.

http://www.openairinterface.org/?page_id=25

[5] Wiki principal del proyecto OAI.

https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/

[6] Libro LTE: Nuevas tendencias en comunicaciones móviles.

http://www.fundacionvodafone.es/sites/default/files/libro_lte.pdf

[7] LTE Network Architecture

https://www.netmanias.com/en/?m=view&id=techdocs&no=5904`

[8] Guión para la implementación de escenarios.

https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/HowToConnectOAIENBWithOAIUEWithS1Interface

[9] OpenAirInterface Alliance, Abril 27, 2017

http://www.openairinterface.org/docs/workshop/3_OAI_Workshop_20170427/training/OAI_basics_kaltenbe_2017.pdf

[10] Tracking Area (TA) and Tracking Area Update (TAU), Agosto 30, 2013 | Dr. Michelle M. Do

https://www.netmanias.com/en/?m=view&id=blog&no=5930

[11] Enciclopedia LTE.

https://sites.google.com/site/lteencyclopedia/lte-network-infrastructure-and-elements

[12] Estudi experimental d'una xarxa LTE amb la plataforma OpenAirInterface (OAI), Mayo 2018 | David Sariol Rustarazo
https://upcommons.upc.edu/bitstream/handle/2117/117764/Memoria_TFG_David%20Sariol.pdf?sequence=1&isAllowed=y

[13] LTE Tracking Area Update Call Flow Procedure, Junio 22, 2017 | Prashant Panigrahi

http://www.3glteinfo.com/lte-tracking-area-update-call-flow-procedure/

[14] Jesús Ramón Pérez,

Asignatura: Comunicaciones móviles e inalámbricas

Departamento de Ingeniería de comunicaciones, Universidad de Cantabria

[15] EMM Procedure 1, Enero 16, 2014 | Netmania

https://www.netmanias.com/en/?m=view&id=techdocs&no=10441

[16] Tarjertas USIM

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

[17] Prototipo de una estación base 4G usando Open Air Interface, Septiembre de 2017 **|** Francisco García Espigares

http://wpd.ugr.es/~jorgenavarro/thesis/2017_TFG_FranciscoGarciaEspigares.pdf

[18] Prototype Implementation of a 5G Group-Based Authentication and Key Agreement Protocol, Diciembre 12, 2016 **|** Markus Ahlström

http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8895975&fileOId=8895979

ANEXO:

ANEXO A: Log de UE en OAI E PC + OAI eNB + OAI UE

```
[RRC][I][rrc_ue_generate_RRCConnectionRequest] [UE 0] : Frame 97, Logical Channel UL-CCCH
(SRB0), Generating RRCConnectionRequest (bytes 6, eNB 0)
[PHY][I][ue_pdcch_procedures] [UE  0] subframe 5: Found rnti ffff, format 1A, dci_cnt 0
[PHY][I][ue_pdcch_procedures] [UE  0] Frame 98, subframe 5 : Generate UE DLSCH SI_RNTI format 1A
[PHY][I][ue_pdcch_procedures] [UE  0] subframe 5: Found rnti ffff, format 1A, dci_cnt 0
[PHY][I][ue_pdcch_procedures] [UE  0] Frame 100, subframe 5 : Generate UE DLSCH SI_RNTI format
1A
[PHY][I][ue_pdcch_procedures] [UE  0] subframe 5: Found rnti ffff, format 1A, dci_cnt 0
[PHY][I][ue_pdcch_procedures] [UE  0] Frame 102, subframe 5 : Generate UE DLSCH SI_RNTI format
1A
[PHY][I][lte_adjust_synch] [UE0] Sending synch status to higher layers
[PHY][I][ue_prach_procedures] mode 0
[PHY][I][ue_prach_procedures] [UE  0][RAPROC] Frame 106, Subframe 1 : Generating PRACH, preamble
28,PL 46,  P0_PRACH -62, TARGET_RECEIVED_POWER -108 dBm, PRACH TDD Resource index 0, RA-RNTI 2
[PHY][I][ue_prach_procedures] [UE  0][RAPROC] Frame 106, subframe 1: Generating PRACH (eNB 0)
preamble index 28 for UL, TX power -62 dBm (PL 46 dB), l3msg
[MAC][I][ue_scheduler] Received RRC_MAC_CCCH_DATA_REQ from TASK_RRC_UE: instance 0, frameP 106,
eNB_index 0
[PHY][I][ue_pdcch_procedures] [UE  0] subframe 5: Found rnti ffff, format 1A, dci_cnt 0
[PHY][I][ue_pdcch_procedures] [UE  0] Frame 106, subframe 5 : Generate UE DLSCH SI_RNTI format
1A
[MAC][I][ue_process_rar] [eNB 0][RAPROC] Frame 123 Received RAR (5b|00.20.35.4c.67.be) for
preamble 27/27
[PHY][I][process_timing_advance_rar] [UE 0] AbsoluteSubFrame 123.0, received (rar)
timing_advance 8
[PHY][I][ue_ulsch_uespec_procedures] [UE  0][RAPROC] Frame 123, Subframe 6 Generating
(RRCConnectionRequest) Msg3 (nb_rb 2, first_rb 1, round 0, rvidx 0) Msg3:
20.6.1f|5f.5a.55.a8.f5.26
[PHY][I][pusch_power_cntl] [UE  0][RAPROC] AbsSubframe 123.6: Msg3 (2 PRBs) Po_PUSCH -42 dBm (-
10800,301,100*PL=4500,0,1800)
[MAC][I][ue_scheduler] Frame 123: Contention resolution timer 0/48
[MAC][I][ue_scheduler] Frame 123: Contention resolution timer 1/48
[MAC][I][ue_scheduler] Frame 123: Contention resolution timer 2/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 3/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 4/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 5/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 6/48
[PHY][I][rx_phich] [UE  0][PUSCH 4][RAPROC] Frame 124 subframe 0 Msg3 PHICH, received NAK (1855)
nseq 1, ngroup 0
```

```
[PHY][I][rx_phich] [PUSCH 4] AbsSubframe 124.0: f_pusch (ACC) 21, adjusting by 3 (TPC 3)
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 7/48
[PHY][I][pusch_power_cntl] [UE  0][RAPROC] AbsSubframe 124.4: Msg3 (2 PRBs) Po_PUSCH -39 dBm (-
10800,301,100*PL=4500,0,2100)
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 8/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 9/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 10/48
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 11/48
[PHY][I][ue_pdcch_procedures] [UE  0] subframe 5: Found rnti ffff, format 1A, dci_cnt 0
[PHY][I][ue_pdcch_procedures] [UE  0] Frame 124, subframe 5 : Generate UE DLSCH SI_RNTI format
1A
[MAC][I][ue_scheduler] Frame 124: Contention resolution timer 12/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 13/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 14/48
[PHY][I][rx_phich] [UE  0][PUSCH 4][RAPROC] Frame 124 subframe 8 Msg3 PHICH, received NAK (1851)
nseq 1, ngroup 0
[PHY][I][rx_phich] [PUSCH 4] AbsSubframe 124.8: f_pusch (ACC) 24, adjusting by 3 (TPC 3)
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 15/48
[PHY][I][pusch_power_cntl] [UE  0][RAPROC] AbsSubframe 125.2: Msg3 (2 PRBs) Po_PUSCH -35 dBm (-
10800,301,100*PL=4600,0,2400)
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 16/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 17/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 18/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 19/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 20/48
[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 21/48


[MAC][I][ue_scheduler] Frame 125: Contention resolution timer 22/48
[PHY][I][rx_phich] [UE  0][PUSCH 4][RAPROC] Frame 125 subframe 6 Msg3 PHICH, received NAK (1863)
nseq 1, ngroup 0
[PHY][I][rx_phich] [PUSCH 4] AbsSubframe 125.6: f_pusch (ACC) 27, adjusting by 3 (TPC 3)
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 23/48
[PHY][I][pusch_power_cntl] [UE  0][RAPROC] AbsSubframe 126.0: Msg3 (2 PRBs) Po_PUSCH -32 dBm (-
10800,301,100*PL=4600,0,2700)
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 24/48
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 25/48
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 26/48
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 27/48
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 28/48
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 29/48
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 30/48
[PHY][I][rx_phich] [UE  0][PUSCH 4][RAPROC] Frame 126 subframe 4 Msg3 PHICH, received ACK (-25)
nseq 1, ngroup 0
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 31/48
[PHY][I][ue_pdcch_procedures] [UE  0] subframe 5: Found rnti ffff, format 1A, dci_cnt 0
[PHY][I][ue_pdcch_procedures] [UE  0] Frame 126, subframe 5 : Generate UE DLSCH SI_RNTI format
1A
[MAC][I][ue_scheduler] Frame 126: Contention resolution timer 32/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 33/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 34/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 35/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 36/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 37/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 38/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 39/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 40/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 41/48
[MAC][I][ue_scheduler] Frame 127: Contention resolution timer 42/48
[MAC][I][ue_scheduler] Frame 128: Contention resolution timer 43/48
[MAC][I][ue_scheduler] Frame 128: Contention resolution timer 44/48
[MAC][I][ue_scheduler] Frame 128: Contention resolution timer 45/48
[MAC][I][ue_scheduler] Frame 128: Contention resolution timer 46/48
[MAC][I][ue_scheduler] Frame 128: Contention resolution timer 47/48
[MAC][E][ue_scheduler] Module id 0 Contention resolution timer expired, RA failed
[PHY][E][ra_failed] [UE 0] Random-access procedure fails, going back to PRACH, setting SIStatus
= 0, discard temporary C-RNTI and State RRC_IDLE
```

ANEXO B: Log de eNB en OAI EPC + OAI eN B + OAI UE

```
[RRC][I][FRAME 00000][eNB][MOD 00][RNTI bcac] [RAPROC] Logical Channel DL-CCCH, Generating
RRCConnectionSetup (bytes 25)
[RRC][I][FRAME 00000][eNB][MOD 00][RNTI bcac]CALLING RLC CONFIG SRB1 (rbid 1)
[PDCP][N][FRAME 00000][eNB][MOD 00][RNTI bcac][SRB 01]  Action ADD  LCID 1 (SRB id 1) configured
with SN size 5 bits and RLC AM
[RLC][I][FRAME 00000][eNB][MOD 00][RNTI bcac] [SRB 1] rrc_rlc_add_rlc  SRB
[RLC][I][FRAME 00000][eNB][MOD 00][RNTI bcac][SRB AM 01][CONFIGURE] max_retx_threshold 4
poll_pdu 4 poll_byte 65535 t_poll_retransmit 80 t_reordering 35 t_status_prohibit 0
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 967 subfarme 7, UE 0: not configured, skipping UE
scheduling
[PHY][I][eNB 0] Sent physicalConfigDedicated=0x7fb564000910 for UE 0
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 967 subfarme 8, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 967 subfarme 9, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 968, subframeP 0: Generating Msg4 with RRC
Piggyback (RA proc 0, RNTI bcac)
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 968 subframeP 0 Msg4 : TBS 41, sdu_len 25,
msg4_header 8, msg4_padding 0, msg4_post_padding 7
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 968 subfarme 0, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 968 subfarme 1, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 968 subfarme 2, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 968 subfarme 3, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 968 subfarme 4, UE 0: not configured, skipping UE
scheduling
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 968, subframeP 5: Checking if Msg4 was
acknowledged:
MAC: msg4 acknowledged for rnti bcac fsf 968/5, let's configure it
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 968, subframeP 5 : Msg4 acknowledged
[MAC][I][schedule_ulsch_rnti] [eNB 0][PUSCH 2/bcac] CC_id 0 Frame 968 subframeP 6 Scheduled UE 0
(mcs 9, first rb 1, nb_rb 3, rb_table_index 2, TBS 57, harq_pid 2)
[MAC][I][schedule_ulsch_rnti] [eNB 0][PUSCH 2/bcac] CC_id 0 Frame 971 subframeP 8 Scheduled UE 0
(mcs 9, first rb 1, nb_rb 3, rb_table_index 2, TBS 57, harq_pid 2)
[MAC][I][schedule_ulsch_rnti] [eNB 0][PUSCH 2/bcac] CC_id 0 Frame 975 subframeP 0 Scheduled UE 0
(mcs 9, first rb 1, nb_rb 3, rb_table_index 2, TBS 57, harq_pid 2)
[RRC][I]Removing UE bcac instance
[RRC][W][eNB 0] Removing UE RNTI bcac
MAC: remove UE 0 rnti bcac
[MAC][I][rrc_mac_remove_ue] Removing UE 0 from Primary CC_id 0 (rnti bcac)
[S1AP][W][s1ap_ue_context_release_req] Failed to find ue context associated with eNB ue s1ap id:
0
[S1AP][E][s1ap_eNB_task] Failed to find ue context associated with eNB ue s1ap id: 0
[SCTP][I][sctp_eNB_read_from_socket] An error occured during read
[SCTP][E][sctp_eNB_read_from_socket] sctp_recvmsg (fd 42, len -1 ): Connection timed out:110
```