UNIVERSITAT POLITÈCNICA DE VALÈNCIA

TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

IMPLEMENTACIÓN DEL NÚCLEO DE RED LTE/5G VIRTUALIZADO

Arturo Mrozowski Handzel

Tutor: Jose Francisco Monserrat del Río

Cotutor: Sergio Pastor Tur

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2019-20

Valencia, septiembre de 2020

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/ CAMPUS VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS Gestión ambiental verificada

Resumen

Entre los numerosos avances tecnológicos que guían el desarrollo de las redes 5G, uno de los más destacados guarda relación con la arquitectura del núcleo de red. La virtualización de las funciones de red (NFV) y las redes definidas por software (SDN) son los modelos cuya aplicación permite llevar a cabo despliegues económicos de núcleos de red modulares y fácilmente escalables que desbloqueen el verdadero potencial de la nueva generación de comunicaciones móviles. Este nuevo paradigma es también la clave para lograr establecer un ecosistema abierto e interoperable de redes móviles.

Este Trabajo Final de Grado (TFG) se centra en la implementación de núcleos de red LTE y 5G funcionales sobre equipos de propósito general mediante las plataformas de código abierto *OpenAirInterface* y *Open5GCore*, con el objetivo de validarlos para su despliegue en una red privada. Se analizan las posibilidades de ambas soluciones y se desarrolla el procedimiento seguido para su configuración. Tras ello, se diseña un escenario para comparar sus prestaciones, integrándolos con una red de acceso radio para conectar dispositivos móviles comerciales y sometiéndolos a una serie de pruebas mediante la herramienta de evaluación de núcleos de red *Landslide*.

Palabras clave: núcleo de red, 5G, NSA, NFV, *OAI*, *Open5GCore*

Resum

Entre els nombrosos avanços tecnològics que guien el desenvolupament de les xarxes 5G, un dels més destacats guarda relació amb l'arquitectura del nucli. La virtualització de les funcions de xarxa (NFV) i les xarxes definides per software (SDN) són els models l'aplicació dels quals permet dur a terme desplegaments econòmics de nuclis modulars i fàcilment escalables que desbloquegen el vertader potencial de la nova generació de comunicacions mòbils. Aquest nou paradigma és també la clau per aconseguir establir un ecosistema obert i interoperable de xarxes mòbils.

Aquest Treball Final de Grau (TFG) se centra en la implementació de nuclis LTE i 5G funcionals sobre equips de propòsit general mitjançant les plataformes de codi obert *OpenAirInterface* i *Open5GCore*, amb l'objectiu de validar-los per al seu desplegament en una xarxa privada. S'analitzen les possibilitats de les dues solucions i es desenvolupa el procediment seguit per a la seua configuració. A continuació, es dissenya un escenari per comparar les seues prestacions, integrant-los amb una xarxa d'accés ràdio per connectar dispositius mòbils comercials i sotmetent-los a una sèrie de proves mitjançant l'eina d'avaluació de nuclis *Landslide*.

Paraules clau: nucli 5G, NSA, NFV, *OAI*, *Open5GCore*

Abstract

Among the many technological advances that guide the development of 5G networks, one of the most prominent relates to the architecture of the network core. Network Functions Virtualization (NFV) and Software-Defined networking (SDN) are the models whose application enables cost-effective deployments of modular and easily scalable network cores that unlock the true potential of the new generation of mobile communications. This new paradigm is also the key to establishing an open and interoperable ecosystem of mobile networks.

This Bachelor's Thesis focuses on the implementation of functional LTE and 5G network cores on general-purpose equipment using the open source platforms *OpenAirInterface* and *Open5GCore*, aiming to validate them for their deployment in a private network. The possibilities of both solutions are analyzed, and the procedure followed for their configuration is developed. After that, a scenario is designed to compare their performance, integrating them with a radio access network to connect commercial mobile devices and subjecting them to a series of tests using the *Landslide* network core evaluation tool.

Keywords: core network, 5G, NSA, NFV, *OAI*, *Open5GCore*

Índice general

Capítulo 1

Introducción

1.1. Contexto

Las Tecnologías de la Información y la Comunicación (TIC) desempeñan un rol fundamental en la actualidad. Siendo la comunicación una necesidad inherente al ser humano y sujeta a un constante proceso de evolución, resulta difícil situar el origen de las TIC, dado que por la propia definición del término *tecnología* pueden considerarse un concepto dinámico [1].

Pese a esta variación en el tiempo, la aparición de sistemas de telecomunicaciones desarrollados especialmente a partir de la segunda mitad del siglo XX y su generalización durante dicho período ha supuesto una verdadera revolución de las sociedades modernas. Es innegable el impacto que las TIC han provocado en todos los ámbitos, desde la economía, al tratarse de un elemento clave en la transformación de los procesos productivos y de consumo, hasta las relaciones sociales, sin olvidar el desarrollo tecnológico implícito.

Dentro de las redes de telecomunicaciones actuales, las celulares son de suma importancia en un mundo caracterizado por la interconexión y la movilidad tanto de capitales como de personas. Su relevancia es tal que, sobre todo en los países desarrollados, es inconcebible la ausencia de comunicaciones móviles en la vida diaria, un hecho que se ha acrecentado en la última década con la aparición del *smartphone* [2].

Los dispositivos móviles disponibles en el mercado han alcanzado una potencia y un nivel de complejidad tan elevados que con ellos es posible realizar multitud de tareas para las que antes se necesitaban varios dispositivos. De hecho, aquellas primeras generaciones de dispositivos pensados para poder realizar llamadas sin depender de la red telefónica fija han dado paso a verdaderos ordenadores en miniatura que, sin perder su cometido original, ponen el foco en el acceso a los múltiples servicios disponibles a través de Internet.

Paralelamente a este cambio paradigmático de la funcionalidad de los dispositivos móviles, se ha producido una evolución de los estándares de las redes de comunicaciones móviles, con el objetivo último de adaptarlas a la nueva realidad y prestar un servicio de comunicaciones ubicuo [3].

En este sentido, no se puede obviar que los progresos en el campo de las comunicaciones móviles son fruto de las exigencias de los consumidores. Su demanda cambiante obliga a

diseñar las redes móviles del futuro en base a la misma, tanto en términos de dispositivos conectados como de servicios ofertados [4]. Esta afirmación puede comprobarse haciendo un repaso por las generaciones de redes celulares inalámbricas de las últimas décadas [5], desde la aparición de la primera generación (1G), con tecnología analógica diseñada únicamente para realizar llamadas de voz, seguida de la segunda generación (2G), que introduce tecnología digital. Después de ellas, la tercera generación (3G) se caracteriza por el soporte multimedia junto a mayores tasas de transmisión de datos. A continuación nace la cuarta generación (4G), que supone una primera integración de las redes de banda ancha móvil y fija. Tecnologías como LTE (*Long Term Evolution*) implementan un sistema de red basado en *Internet Protocol* (IP) de extremo a extremo, pensado para dar respuesta a las limitaciones de 3G. Mejora la calidad de servicio (QoS), las tasas de transmisión de datos y reduce el coste de los recursos.

Actualmente, las nuevas demandas de la sociedad están planteando nuevos retos para las redes móviles y, del mismo modo que ha venido ocurriendo hasta ahora, el sector de las telecomunicaciones dará respuesta a las mismas con una nueva generación, la 5G. Las redes de quinta generación prometen suponer una verdadera revolución en el ámbito de las comunicaciones móviles, pues al permitir la interoperabilidad con otras redes móviles e inalámbricas materializarán un auténtico mundo inalámbrico [5].

Los consumidores de hoy requieren redes móviles que ofrezcan mayores velocidades de descarga, mayor fiabilidad y una experiencia de usuario mejorada [4]. Asimismo, cabe esperar que en los próximos años se produzca la generalización de aplicaciones como los vehículos autónomos, las ciudades y hogares inteligentes, la realidad virtual y aumentada, los servicios médicos a distancia o el Internet de las cosas (IoT).

En definitiva, en los próximos años se desplegará una red nunca vista hasta el momento, que conectará personas, máquinas e incluso ciudades y sistemas de transporte. Cuando sea una realidad, las nuevas redes móviles que se despliegen para dar servicio a estas nuevas aplicaciones deberán poder hacer frente a un volumen de tráfico de datos inmenso a una velocidad mucho mayor a la actual, permitir la conexión de dispositivos de forma masiva sin que ello repercuta en la fiabilidad y reducir al mínimo el retardo o latencia con que se entregarán los datos, todo ello al mismo tiempo.

No obstante, respaldar este enorme y rápido incremento en la cantidad de datos transmitidos y la conectividad supone un problema para las redes 4G LTE actuales. Pese a que se sigue trabajando en diferentes líneas de investigación para hacer frente a esta explosión de tráfico mediante LTE, velocidades teóricas máximas de 150 Mbps o un máximo de 600 usuarios conectados por celda no parecen suficientes para que las comunicaciones de banda ancha móvil actuales ofrezcan un servicio adecuado a las nuevas necesidades [6]. En este contexto, el despliegue de redes móviles 5G se convierte en un pilar fundamental para el mundo que viene.

El reciente informe de Ericsson sobre movilidad [7] permite observar el crecimiento de aquellos factores que justifican el necesario desarrollo de la 5G. Dicho informe contiene previsiones en línea con las realizadas por otras empresas del sector o instituciones.

Así, en la Figura 1 se puede comprobar el aumento esperado en el número de suscripciones a redes móviles. Con alrededor de 8000 millones de suscripciones en la actualidad, se esperan cerca de 9000 millones en 2025, de los cuales unos 2800 millones serán sus-

cripciones de redes 5G. Este número se alcanzará a medida que se vayan produciendo a gran escala los lanzamientos comerciales de las redes por parte de los operadores, previstos para cuando estén completadas las especificaciones IMT-2020 de la Unión Internacional de Telecomunicaciones (UIT) sobre 5G, hacia finales del año 2020.

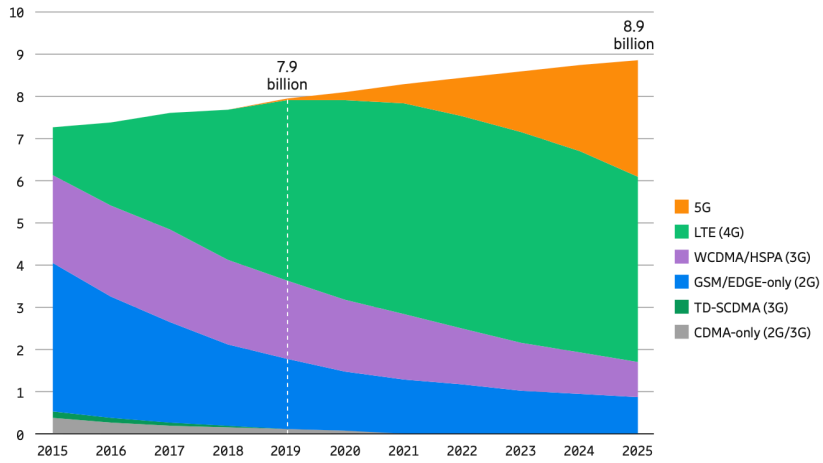

Figura 1: Suscripciones móviles por tecnología (en miles de millones) [7]

Por otra parte, en la Figura 2 puede observarse el crecimiento exponencial que viene experimentando el tráfico de datos en las redes móviles de todo el mundo hasta el momento, tendencia que se mantendrá en los años venideros.

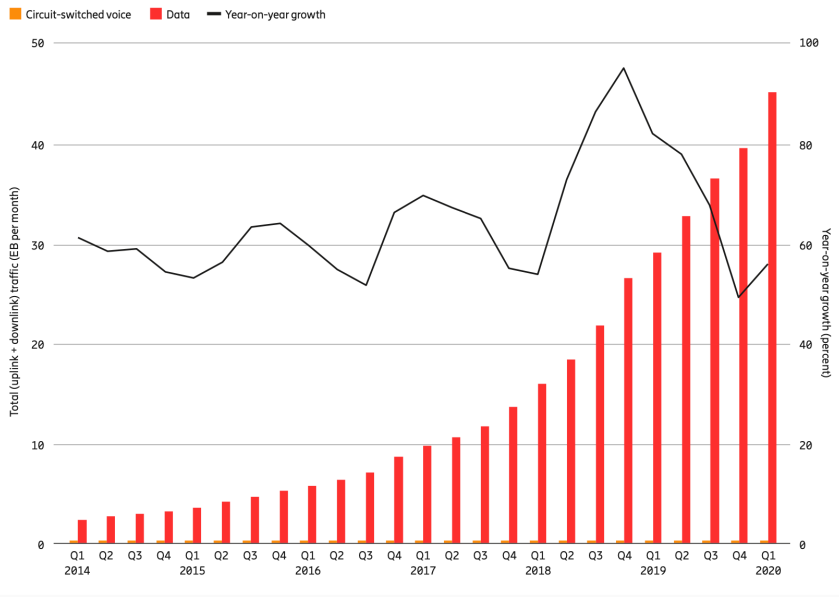

Figura 2: Tráfico global de datos en las redes móviles y crecimiento anual (EB/mes) [7]

1.2. Motivación

Hasta hace unos años, conceptos como IoT, VR (realidad virtual), comunicaciones M2M (máquina a máquina) o V2X (vehículo a todo), domótica y *smart cities* parecían ideas del futuro. Sin embargo, el desarrollo tecnológico es imparable y en estos momentos son ya una realidad.

Durante la tercera década del siglo XXI se espera un crecimiento notorio de sistemas y dispositivos que harán uso de las numerosas posibilidades que estas nuevas tecnologías ofrecen. El mundo se encuentra, pues, a las puertas de una nueva era que se caracterizará por la automatización y la hiperconectividad.

Para completar con éxito la transformación descrita, las redes de banda ancha inalámbricas, y más específicamente las móviles, serán fundamentales, en tanto que constituirán el soporte de toda la red de dispositivos inteligentes. Por tanto, si a día de hoy la relevancia que las comunicaciones móviles tienen en el conjunto de las TIC es manifiesta, lejos de disminuir, el papel de las mismas será todavía más importante en el futuro.

Dado que muchas de las tecnologías presentadas se encuentran en fase de investigación y desarrollo, resulta especialmente interesante disponer de redes celulares desplegadas en entornos de laboratorio y configuradas según las necesidades de cada caso, de modo que se evite la dependencia de redes comerciales. Este proyecto se enfoca en el núcleo, uno de los elementos críticos de una red móvil.

Si bien la tecnología LTE ha alcanzado un considerable grado de madurez, al datar de 2008 el estándar completado por el grupo de asociaciones de telecomunicaciones 3GPP (*3rd Generation Partnership Project*) en la llamada *Release* 8, esta sigue siendo uno de los elementos clave en el desarrollo de las redes 5G.

La *Release* 15 del 3GPP, finalizada en 2018, desarrolla la tecnología conocida como *New Radio* (NR) para la 5G, y ha sido adoptada en el marco de la hoja de ruta IMT-2020 de la UIT como el estándar para la nueva generación de redes móviles. Dichas especificaciones contemplan dos posibles arquitecturas para las redes 5G: NSA (*Non-Stand-Alone*) y SA (*Stand-Alone*). Ambas suponen el despliegue de una red de acceso radio (RAN) con estaciones base gNB (*gNodeB*) basadas en NR, pero NSA sigue dependiendo del núcleo de red, denominado (*Evolved Packet Core*), y las estaciones base eNB (*eNodeB*) de LTE.

Debido a que el EPC es una pieza clave en el despliegue de las primeras redes 5G, las NSA, el presente proyecto se centra tanto en la arquitectura del núcleo de red 4G como la de 5G (5GC, *5G Core*), que constituirá la novedad de la 5G SA. Además, la quinta generación viene acompañada de nuevos enfoques para la implementación de los elementos de la red. Se apuesta de forma definitiva por las posibilidades que ofrece la virtualización frente al predominio tradicional de equipos físicos propietarios diseñados para un fin específico [8].

La respuesta a los retos que plantea la nueva generación de redes móviles pasa por arquitecturas basadas en las redes definidas por *software* (SDN) y los equipos radio definidos por *software* (SDR) [9]. Se persigue facilitar la escalabilidad de las redes y reducir los costes. El uso de estas tecnologías posibilita desplegar redes celulares operativas con equipamiento de bajo coste, en comparación con las redes tradicionales.

1.3. Objetivos

El objetivo principal del presente Trabajo Final de Grado, enmarcado en el proyecto Valencia Campus 5G, es implementar sendos núcleos de red LTE y 5G funcionales de código abierto sobre equipos de propósito general y estudiar sus prestaciones. Con ello se podrá proceder al despliegue de redes móviles privadas de estas características, que cumplan con las especificaciones recogidas en los estándares del 3GPP, en el Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM) de la Universitat Politècnica de València (UPV).

Para ello, se identifican los siguientes objetivos específicos que deben abordarse:

- Estudio de la arquitectura del núcleo de red 5G y su evolución respecto a la de 4G.
- Instalación y configuración del *hardware* y *software* necesario en los equipos en los que se desplegarán los núcleos de red.
- Implementación de núcleos de red LTE con soporte para redes 5G de tipo NSA mediante las plataformas de código abierto *OpenAirInterface* (OAI) y *Open5GCore*, así como de un núcleo 5G preparado para redes de tipo SA a través de esta última.
- Validación de las soluciones propuestas mediante la integración con la red de acceso radio (RAN) 4G y 5G, tanto emulada como desplegada físicamente utilizando un equipo SDR y dispositivos móviles comerciales para tal fin.
- Diseño de una serie de pruebas para analizar y comparar el rendimiento de los distintos núcleos de red implementados en determinados supuestos a través de la herramienta *Spirent Landslide*.

1.4. Estructura de la memoria

Este apartado pretende servir de guía al lector, exponiendo los contenidos que aborda la presente memoria y su organización, para facilitar la búsqueda de información a lo largo de la misma. El resto de capítulos que componen el documento son:

Capítulo 2. Estado del arte

En este capítulo se presenta la arquitectura de las redes móviles LTE y 5G y, en particular, de sus respectivos núcleos de red, revisando el marco tecnológico que posibilita el desarrollo de arquitecturas de núcleos de red móvil de nueva generación, además de comentar la convergencia del sector hacia un modelo *open source*.

Capítulo 3. Consideraciones previas

Este capítulo constituye una descripción general de la solución propuesta, listando los requisitos que deben satisfacer los equipos utilizados para desarrollar dicha solución.

Capítulo 4. Despliegue del núcleo de red de *OpenAirInterface*

Este capítulo desarrolla detalladamente el proceso seguido para el despliegue de un núcleo de red LTE implementado mediante el *software OpenAirInterface*, incluyendo las características de la plataforma y su instalación y configuración.

Capítulo 5. Despliegue de *Open5GCore*

En este capítulo se describen los pasos seguidos para desplegar la implementación *software* de núcleos de red LTE y 5G *Open5GCore*, explicando las particularidades de esta solución y el modo de instalarla y configurarla.

Capítulo 6. Diseño del escenario de pruebas

Este capitulo explica el procedimiento seguido para evaluar las prestaciones de los núcleos de red implementados, consistente en integrar una red de acceso radio para conectar dispositivos móviles y realizar pruebas con la herramienta comercial *Landslide*.

Capítulo 7. Análisis de resultados

En este capítulo se exponen los resultados obtenidos en las diferentes pruebas desarrolladas para determinar el rendimiento de los núcleos de red desplegados, estableciendo una comparativa entre ambos.

Capítulo 8. Conclusiones y propuestas de trabajo futuro

En este capítulo final se presentan las conclusiones alcanzadas a partir de la realización del trabajo y los resultados obtenidos, ofreciendo una valoración global del proyecto y sugiriendo líneas futuras de desarrollo.

Capítulo 2

Estado del arte

Este capítulo constituye la conceptualización de la arquitectura de las redes móviles de cuarta y quinta generación, con especial atención a los núcleos de red que forman parte de ellas. A continuación, se revisan las principales tecnologías que guían el desarrollo de la arquitectura de las redes móviles de nueva generación. En relación con ello, se discuten distintas soluciones comerciales y de código abierto disponibles en el mercado que hacen uso de estas novedades tecnológicas para llevar a cabo la implementación de los núcleos de red de LTE y 5G.

2.1. Arquitectura de redes móviles LTE

En esta sección se introduce al lector la arquitectura de las redes móviles de cuarta generación (4G) desplegadas al amparo del estándar LTE (*Long Term Evolution*) del 3GPP, profundizando en el núcleo de red, el EPC (*Evolved Packet Core*). Puede consultarse [10] y [11] para ampliar la información.

LTE es el estándar de las comunicaciones móviles de cuarta generación (4G). Desarrollado por el 3GPP y documentado en su *Release* 8, el estándar se completó en 2008. LTE supuso una evolución de las tecnologías de comunicaciones móviles disponibles hasta ese momento. La premisa era buscar una simplificación de la red móvil en torno a dos partes bien diferenciadas: un núcleo de red, el EPC, y una red de acceso radio (RAN) que conecte a los usuarios al mismo.

Para la definición de ambas partes de la red se tomó como base el desarrollo de las redes de tercera generación (3G). La RAN se renovó adoptando las nuevas tecnologías de acceso radio para dar lugar a la llamada E-UTRAN (*Evolved Universal Terrestrial Access Network*). Por su parte, el EPC abandonó el soporte a la conmutación de circuitos para convertirse en el primer núcleo de red móvil totalmente basado en la arquitectura TCP/IP.

Asimismo, otro de los objetivos fijados para el diseño de la arquitectura LTE fue optimizar el manejo del tráfico de datos desde y hacia los usuarios. Para ello se optó por una arquitectura plana. En este contexto, plana hace referencia a reducir al mínimo el número de nodos implicados en el proceso de conmutación de paquetes. Esta nueva arquitectura permitió despliegues más económicos y escalables y pudo hacer frente a la

creciente demanda de los usuarios.

Así, la red de acceso radio está centralizada en un único elemento, el eNB (*eNodeB*, al mismo tiempo que el número de entidades del EPC se ve reducido a un grupo de nodos que centralizan las funciones esenciales de un núcleo de red.

2.1.1. Componentes del EPC

Una de las decisiones iniciales del diseño de la arquitectura del EPC fue separar la gestión de la señalización de control del tráfico de datos generado por los usuarios. Esta separación se realizó principalmente para optimizar la escalabilidad de las redes, en tanto que el volumen de mensajes de señalización aumenta con el número de suscriptores móviles, pero la cantidad de paquetes de datos guarda relación con la demanda y las nuevas aplicaciones y servicios que van surgiendo.

La Figura 3 muestra la arquitectura elemental del EPC de una red LTE. Los componentes que se incluyen son los que, según las especificaciones del 3GPP, debe incluir un núcleo de red para cumplir el estándar LTE. Sin estos componentes, no podría conseguirse la funcionalidad básica del núcleo de una red móvil. Además, pueden distinguirse las entidades que gestionan mensajes de señalización (líneas punteadas) de las que manejan el tráfico de los usuarios.

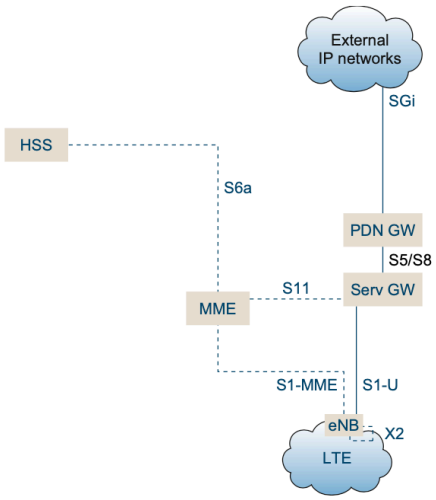

Figura 3: Arquitectura básica del EPC en una red LTE [10]

MME

El MME, *Mobility Management Entity* o Entidad de Gestión de Movilidad, es el elemento que centraliza la señalización LTE del plano de control relacionada con las suscripciones y sesiones de los usuarios.

El MME implementa entre sus funcionalidades procedimientos de seguridad a través de la negociación de algoritmos de cifrado y de protección de la identidad para la autenticación del usuario. También gestiona la sesión establecida entre el terminal de usuario (UE) y la red, siendo la entidad responsable de la señalización utilizada para negociar un contexto

de transmisión de paquetes de datos y los parámetros asociados al mismo, como la calidad de servicio (QoS).

Los distintos eNB de una red se conectan al MME del núcleo de red a través de la interfaz S1-MME, también identificada como S1-C, mientras que la conexión al S-GW se realiza mediante la interfaz S11, siendo el MME la entidad que selecciona los nodos *Gateway* (puerta de enlace) asociados al usuario.

HSS

El HSS, *Home Subscriber Service* o Servidor de Abonados Domésticos, es una base de datos central que se almacena en un solo nodo y que da servicio a las entidades de control del núcleo de red que gestionan el tráfico de datos de los usuarios.

El servidor almacena los datos que identifican a los suscriptores a los que el operador de red da servicio. Los parámetros principales son el IMSI (*International Mobile Subscriber Identity* y el MSISDN (*Mobile Station Integrated Services Digital Network*). También almacena información sobre los servicios contratados y Qos.

Por último, el HSS es el nodo en el que se genera la información de seguridad necesaria para controlar el acceso a la red y llevar a cabo el procedimiento mutuo de autenticación y autorización entre los UE y la red [12]. La interfaz que utiliza para comunicar al MME estos datos es la S6a.

S-GW

El S-GW (*Serving Gateway*) es uno de los nodos que gestiona el tráfico de datos de los usuarios. Constituye el punto de interconexión entre el plano de conmutación de paquetes y los eNB de la red de acceso radio, utilizándose la interfaz S1-U para el enlace.

El S-GW proporciona un punto de anclaje en la red para la movilidad del UE entre los eNB. Es por tanto, el nodo responsable de gestionar los procesos de movilidad entre celdas (*handover*) con tecnología E-UTRAN.

PDN GW

El PDN GW (*Packet Data Network Gateway*) o P-GW es, al igual que el S-GW, un punto de interconexión, en este caso con redes IP externas (PDN). Es la entidad que gestiona el encaminamiento de los paquetes desde el núcleo de red hacia dichas redes, por ejemplo Internet. El enlace se realiza a través de la interfaz SGi.

Entre las funciones del PDN GW se hallan la asignación de direcciones IP a los usuarios y el control de las reglas definidas por el operador sobre los recursos disponibles para los mismos.

El S-GW y el PDN GW comparten entre ellos la interfaz S5/S8. Esta interfaz, de la que se definen dos variantes, es un caso único dentro del EPC. La interfaz S5 es una interfaz interna de la red del operador, mientras que la interfaz S8 se utiliza cuando el usuario está conectado en itinerancia (*roaming*) a la red de un operador distinto al suyo.

2.1.2. Separación del plano de control y el plano de usuario

Como ya se ha visto, la visión inicial del EPC fue definir una serie de nodos con interfaces exclusivas para la gestión independiente de los mensajes de señalización y de los paquetes de datos, respectivamente. Sin embargo, el crecimiento exponencial del tráfico generado por los usuarios en los últimos años ha acrecentado la necesidad de facilitar todavía más si cabe la escalabilidad de las redes [13].

Para facilitar dicho proceso, el 3GPP definió en la *Release* 14 un elemento que ha resultado clave para los operadores: la separación del plano de control y el plano de usuario (CUPS) de los nodos del EPC. Esta novedad en la arquitectura afecta a aquellos componentes con funciones tanto en la señalización del plano de control como la conmutación de paquetes en el plano de usuario, como el S-GW y el P-GW.

La Figura 4 muestra la arquitectura CUPS con las interfaces que se definen para la conexión de los nuevos nodos [13]. Aparece también el TDF, *Traffic Detection Function* o Función de Detección de Tráfico, que no forma parte del resumen del EPC.

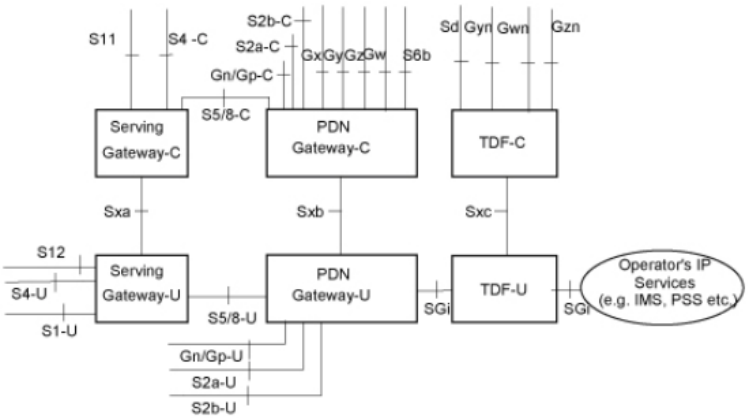

Figura 4: Arquitectura CUPS del EPC [13]

En la arquitectura CUPS, el plano de control de una función puede conectarse a varias funciones de plano de usuario, mientras que el plano de usuario de una función puede ser compartido por varias funciones de plano de control. Para ello se habilitan las interfaces SXa, SXb y SXc que interconectan las funciones del plano de control y el plano de usuario. El SGW-C y el SGW-U en conjunto ofrecen la funcionalidad del S-GW. Lo mismo ocurre con el PGW-C y el PGW-U. La interfaz S5/S8 que conecta ambos nodos en una arquitectura convencional del EPC también se divide en dos.

2.2. Arquitectura de redes móviles 5G

En esta sección se presenta de forma resumida la arquitectura de las redes móviles de quinta generación (5G), poniendo el foco en su núcleo de red, el 5GC (5G *Core*). El lector puede encontrar información más detallada en [14].

Las especificaciones del núcleo de red 5G han sido desarrolladas por el 3GPP en la

Release 15, entre 2018 y 2019. Principalmente, en dichas especificaciones se recoge la arquitectura del núcleo de red en los dos escenarios de redes 5G también definidos en la *Release* 15 (*Stand-Alone* y *Non-Stand-Alone*) y se desarrolla la integración con tecnologías presentadas en este capítulo: la virtualización de redes, el despliegue basado en la nube y el *network slicing*. Posteriormente, en la *Release* 16 completada a principios de 2020 se han incluido varias novedades que abarcan principalmente las aplicaciones que potencia la 5G, como las comunicaciones V2X o IoT.

2.2.1. Escenarios de despliegue de redes 5G

Como se ha apuntado en el párrafo anterior, a día de hoy existen dos configuraciones para el despliegue de redes 5G: SA y NSA. La primera consta de una única tecnología de acceso radio propia de la 5G, NR (*New Radio*), implementada en el gNB (*gNodeB*), y un núcleo de red 5G. Mientras tanto, NSA implementa dos generaciones de redes de acceso radio basadas en LTE y NR conectadas a un EPC. Existen diversas combinaciones de estas tecnologías, aunque los despliegues iniciales de redes 5G han apostado por las dos cuya estandarización está actualmente completada [15]. Dichos escenarios están representados en la Figura 5.

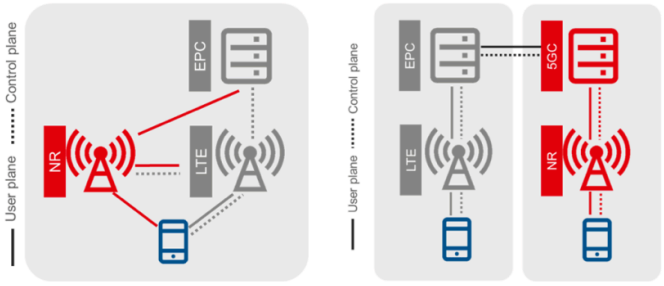

Figura 5: Opciones de despliegue de redes 5G: NSA (izda.) y SA (dcha.) [15]

En un escenario NSA, conocido formalmente como EN-DC (E-UTRAN - NR *Dual Connectivity*), la RAN se compone de eNB que actúan como nodos primarios y gNB que tienen el papel de nodos secundarios. En NSA, los gNB gestionan el tráfico en el plano de usuario, compartiendo parte del mismo con los eNB de los cuales dependen a través de la interfaz X2. Por su parte, el tráfico de señalización entre el núcleo de red LTE y el usuario es competencia de los eNB.

El escenario 5G SA solamente consta de una RAN formada por gNB que se conectan a un núcleo de red 5G, por lo que todo el tráfico generado en la red, tanto de datos como de señalización, es gestionado por dichas estaciones base. La interacción de las redes 5G SA con despliegues 4G LTE se lleva a cabo entre sus respectivos núcleos de red.

Asimismo, en la visión de las redes 5G se definen tres casos de uso, la banda ancha móvil mejorada (eMBB), las comunicaciones ultraconfiables de baja latencia (uRLLC) y las comunicaciones masivas de máquinas (mMTC). De todos ellos, los despliegues de redes 5G actuales, con configuración NSA, solo soportan el primero de ellos. En este sentido puede comprobarse la importancia del núcleo de red 5G como elemento clave en el desbloqueo del potencial de la nueva generación. Por ello, la *Release* 16 abre paso a los casos de uso

restantes.

2.2.2. Funciones de red del 5GC

En comparación con la arquitectura del EPC, la arquitectura del 5G *Core* es al mismo tiempo similar y diferente. Las partes de la red que procesan los paquetes de datos de los usuarios y las que intervienen en la integración con la RAN presentan una estructura parecida a la de la generación anterior. Sin embargo, la parte de la red encargada de la señalización se ha definido de una forma completamente distinta a lo visto en el EPC.

El núcleo de red 5G supone también el cambio de modelo de una red integrada por entidades o nodos conectados a través de interfaces punto a punto a una arquitectura SBA (*Service-Based Architecture*). Estrechamente relacionada con la virtualización de funciones de red, SBA apuesta por este paradigma para el despliegue de los servicios de la red, haciendo que los servicios de una función de red estén disponibles para el resto. En una interacción bajo este modelo se establecen dos roles, el consumidor del servicio y el productor del mismo.

La función principal del núcleo de red es establecer sesiones de forma segura y redireccionar el tráfico desde y hacia los usuarios, dotándoles de conectividad a una red IP. Por tanto, aquellas funciones de red que intervienen en estos procedimientos son las que obligatoriamente deben formar parte de un despliegue para que pueda considerarse un núcleo de red 5G de acuerdo con las especificaciones del 3GPP.

La Figura 6 muestra la arquitectura básica de un 5GC y su integración con la RAN y PDN externas, así como las interfaces utilizadas por las funciones de red para la comunicación entre sí.

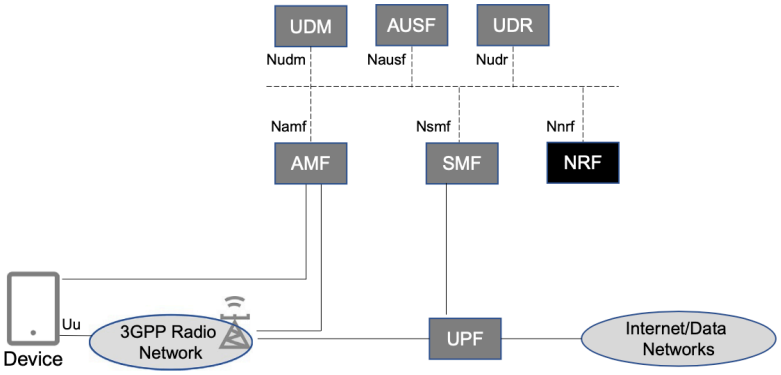

Figura 6: Funciones de red elementales en una red 5G [14]

AMF

El AMF, *Access and Mobility Management Function* o Función de Gestión del Acceso y la Movilidad, es la función principal de control dentro de la red. Interactúa con la RAN y los equipos de usuario (UE) mediante señalización encriptada a través de las interfaces N2 y N1, respectivamente. La mayor parte del flujo de mensajes de señalización pasa por el AMF.

El AMF permite a los dispositivos registrarse, autenticarse y moverse entre las distintas celdas de la red. Sin embargo, no maneja algunas de estas atribuciones por sí mismo, sino que las pide a otras funciones de la red (autenticación) o directamente reenvía los mensajes de señalización a las funciones de red correspondientes (gestión de la sesión). Entre las funciones del AMF también está activar dispositivos de la red que están en modo inactivo.

SMF

El SMF, *Session Management Function* o Función de Gestión de la Sesión, es la función independiente encargada de administración de las sesiones de los usuarios. Ello incluye el establecimiento, la modificación y la liberación de sesiones, así como la asignación de direcciones IP por dispositivo.

El SMF se comunica indirectamente con los usuarios por medio del AMF, que reenvía los mensajes de señalización a través de la interfaz N11 después de procesarlos (por ejemplo, por razones de seguridad).

Además, el SMF interactúa con otras funciones de red, especialmente las Funciones de Plano (UPF), a las que selecciona y controla través de la interfaz N4. Forma parte de este control la configuración del direccionamiento del tráfico y para las distintas sesiones de usuario.

UPF

El UPF, *User Plane Function* o Función del Plano de Usuario, tiene como tarea principal procesar y reenvíar el tráfico del plano de usuario, y se halla bajo el control del SMF.

El UPF es el elemento de conexión del núcleo de red con redes IP externas a través de la interfaz N6 y actúa como un punto de fijación de la IP de los dispositivos hacia redes externas, ocultando la movilidad. Esto significa que cualquier paquete IP cuya dirección de destino sea un dispositivo de la red siempre se podrá encaminar hacia el UPF que sirve al dispositivo, aunque se esté moviendo por la red.

Además, el UPF realiza varios tipos de procesamiento de los datos reenviados. Por un lado, genera informes de uso de tráfico para el SMF, que los incluye luego en sus informes a otras funciones de red. Por otro, puede analizar el contenido de los paquetes de datos y que constituya un elemento más en las decisiones de políticas.

El UPF también se ejecuta en varias políticas de red o de usuario, por ejemplo, la redirección del tráfico, la imposición de diferentes velocidades de conexión o la activación de dispositivos (el UPF almacena los datos dirigidos a dispositivos en estado inactivo al mismo tiempo que los fuerza a volver a conectarse para recibir sus datos).

UDM

El UDM, *Unified Data Management Function* o Función de Gestión de Datos Unificada, representa la base de datos de suscriptores móviles principal. Genera las credenciales de autenticación utilizadas para autenticar los dispositivos que se conectan a la red. También autoriza el acceso a usuarios específicos según la información disponible en la base de datos. Esto permite, por ejemplo, aplicar diferentes normas de acceso a clientes en itinerancia y propios.

El UDM también ejecuta diversas funciones a petición del AMF, con el que se comunica mediante la interfaz N8, así como con el SMF (interfaz N10).

AUSF

El AUSF, *Authentication Server Function* o Función de Servidor de Autenticación, tiene un cometido bastante limitado a la vez que importante. Es el proveedor del servicio de autenticación de los dispositivos, proceso en el que utiliza las credenciales de autenticación generadas por el UDM, al que se conecta mediante la interfaz N13.

NRF

El NRF, *Network Repository Function* o Repositorio de Funciones de Red, es el elemento del núcleo de red que registra las funciones de red productoras de servicios y los servicios que ofrecen para posibilitar la comunicación entre las funciones consumidoras de servicios y ellas.

2.3. Virtualización de funciones de red (NFV)

La virtualización de funciones de red (NFV, *Network Functions Virtualization*) es un conjunto de conceptos propio de las arquitecturas de redes que ofrece una metodología para el diseño, la implementación y el despliegue de redes basada en las tecnologías de virtualización.

Se considera que el marco de referencia de NFV comienza a tomar forma en 2012, cuando un grupo de operadores de telecomunicaciones publica un libro blanco [16] describiendo la visión original de este concepto y los principios en torno a los cuales debe desarrollarse. El llamamiento a otros actores del sector que se realiza al final del documento condujo a la creación de un Grupo de Especificaciones de la Industria sobre la Virtualización de Funciones de Red (ISG NFV) bajo el paraguas del ETSI (*European Telecommunications Standards Institute*).

En [16] se define NFV como la implementación de funciones de red en *software* que puede ejecutarse en un abanico de *hardware* de servidores estándar en la industria, y que pueden ser desplazadas a distintas ubicaciones en la red según se requiera, sin la necesidad de instalar nuevos equipos.

Tradicionalmente, las redes desplegadas por los operadores han estado compuestas por varias plataformas *hardware* de proveedores, asociándose cada elemento de la red a un dispositivo en concreto. Este enfoque conlleva la personalización de los equipos y el desarrollo de un *software* específico para su ejecución en los mismos, minimizando enormemente las posibilidades de reutilización de estas unidades para otras funciones.

Las redes basadas en *hardware* requieren ubicaciones que garanticen el suficiente espacio y potencia para los distintos equipos a instalar, y esta problemática aumenta conforme los operadores lanzan nuevos servicios hasta el punto de no poder desplegarlos por no disponer de espacio físico. Además, la asociación de un equipo a una única función de red da lugar al desaprovechamiento de los primeros.

Como ejemplo de la afirmación anterior, los servidores empresariales no virtualizados desplegados en la actualidad tan solo aprovechan del 10 al 15 % de sus recursos [17]. Pese

a que la potencia de procesamiento del *hardware* actual ha aumentado exponencialmente, estos equipos soportan una única aplicación sobre un único sistema operativo.

Por otro lado, los aparatos diseñados por los proveedores tienen generalmente un ciclo de vida corto y el desarrollo tecnológico actual tiende a acortar todavía más el tiempo que estos dispositivos permanecen operativos. Esto, sumado a su creciente complejidad y la especialización necesaria para integrar estas soluciones en las redes y manejarse con ellas, provoca que los operadores no consigan un retorno de la inversión en estos equipos [16].

Resumiendo, el paradigma de redes basadas en *hardware* parece agotado de cara a los retos que plantean las redes del futuro. En este contexto, NFV es la respuesta a las demandas de los operadores de red. Las compañías que suscriben [16] destacan algunos beneficios de este nuevo modelo, tales como la reducción en el coste de los equipos, una disminución del tiempo transcurrido desde que se diseña un nuevo servicio hasta que se lanza al mercado, debido a que las economías de escala previas no aplican en este caso, la escalabilidad y optimización de las redes en función del tráfico o la reutilización de las máquinas virtuales desplegadas.

Las redes diseñadas mediante funciones virtualizadas también garantizan una mayor operatividad de las redes, puesto que las máquinas virtuales son fácilmente trasladables a otro entorno para su ejecución si surgen problemas, pudiendo completarse esta tarea en cuestión de segundos. En cambio, en el modelo anterior, si una estructura *hardware* falla, en el mejor de los casos sería necesario reiniciar el equipo, pudiendo llegar a tener que sustituirse por otro, dejando a la red desprovista de dicho elemento. Los entornos por los que más apuestan los operadores de telecomunicaciones para el despliegue de NFV son la computación en la nube y los servidores estándar de alto volumen (SHVS, *Standard High Volume Servers*).

2.3.1. Arquitectura de NFV

La Figura 7 muestra los tres componentes principales de la arquitectura de NFV: las funciones de red virtuales (VNF), la infraestructura NFV (NFVI, abajo) y el marco de administración y orquestación (MANO, derecha) [18]. En NFV, las funciones de red convencionales se implementan a través de máquinas virtuales. Las VNF pueden estar compuestas por una o varias máquinas virtuales, dependiendo de si se ha descompuesto la función de red, y están gobernadas por los llamados sistemas de administración de elementos (EMS).

Por otro lado, la infraestructura NFV es el conjunto del *hardware* y *software* en el que se despliegan las VNF. Está formada por los recursos virtuales generados a partir de la división de los recursos *hardware* (computación, almacenamiento y red) por medio de una capa *software* de virtualización. Finalmente, MANO es la parte responsable de la gestión de las VNF y la capa de infraestructura, distinguiéndose tres bloques para ello. También se coordina con sistemas tradicionales de soporte de red.

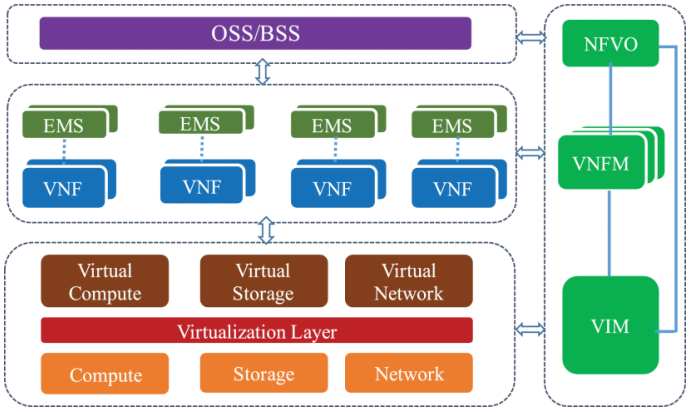

Figura 7: Esquema de la arquitectura de NFV [18]

2.3.2. Núcleos de red móvil basados en NFV y relación con SDN

Uno de los primeros documentos publicados por el ISG NFV es la especificación relativa a los casos de uso de NFV [19]. En esa lista se incluye una de las redes sobre las que los operadores de telecomunicaciones han mostrado un mayor interés en portar a este modelo: los núcleos de red móvil.

El documento describe la virtualización de las entidades que conforman el EPC de LTE y la coexistencia con redes cuyos núcleos no están virtualizados o solo lo están en parte. Se apunta también, como ya se ha comentado anteriormente, a la escalabilidad de elementos concretos de la red, aumentando o disminuyendo los recursos de la NFVI en función de los patrones de tráfico y movilidad de los usuarios.

El uso de NFV para la implementación de núcleos de red permite a los operadores móviles realizar despliegues mucho más rápidos y sencillos al evitar tener que instalar costosos y complicados equipos *hardware* y recurrir a máquinas virtuales sobre nubes, servidores o *switches* de propósito general. De este modo, a los operadores se les presenta la oportunidad de alcanzar un cierto grado de independencia respecto a los proveedores de aparatos.

Además de NFV, otro de los conceptos que ha revolucionado el marco de los núcleos de red móvil es el de redes definidas por *software* (SDN, *Software Defined Networks*). Este paradigma propone una estructura abstracta de las redes, aislando el plano de control del resto de la red y centralizándolo en una o varias entidades de control (controladores SDN), al mismo tiempo que el plano de reenvío de datos se simplifica y se dota a las aplicaciones y servicios de capacidad de gestión de la red.

En [18] se discute la integración de estas tecnologías que, pese a ser diferentes y poner el foco en aspectos diferentes de las redes, son altamente complementarias. NFV provee de elementos virtualizados a las SDN, mientras que SDN ofrece la gestión de la red, entendida como programabilidad de la misma, a las distintas funciones de red virtuales.

Así pues, NFV y SDN se presentan como el futuro de las redes móviles. Su uso, sumado a las posibilidades que sobre todo aporta el *cloud computing*, supone un recorte en la inversión requerida (CAPEX) y los costes de operación (OPEX) de los operadores. También permite variar la topología de la red, acercando determinadas funciones a los usuarios, lo

que repercute en una menor latencia.

Estos paradigmas, que se han ido aplicando gradualmente a las redes LTE desplegadas actualmente, son sin embargo uno de los fundamentos del desarrollo de las redes 5G. El concepto de *slicing* (rebanado) de las redes consiste en prestar servicios de toda índole a través de diferentes *slices* (rebanadas). Cada una de ellas se compone de un conjunto de funciones de red y de los recursos necesarios para ejecutar las mismas, seleccionados de entre toda la infraestructura en función del servicio a prestar. La realización de este concepto, presente desde el principio en la visión de las redes 5G, es posible gracias a NFV y SDN [20]..

2.4. Convergencia hacia redes móviles de código abierto

La apuesta decidida por tecnologías como NFV, SDN o *cloud computing* para el desarrollo de los núcleos de red móvil del futuro ha implicado un reposicionamiento de los actores del sector de las telecomunicaciones. Tanto los fabricantes de equipos como los operadores de red deben trazar nuevas estrategias.

Los proveedores tradicionales de *hardware* para la implementación de núcleos de red abandonaron el enfoque de equipos propietarios y comenzaron a ofrecer *software* licenciado instalado en equipos de propósito general con las arquitecturas de procesadores x86-64 o ARM. El nacimiento de la 5G ha venido acompañado de un nuevo cambio de modelo, priorizando ahora las soluciones basadas en la nube.

Sin embargo, la tendencia en el sector es desarrollar capas *software* que, además de ser plenamente independientes de cualquier tipo de aparatos *hardware* clásicos, sean de código abierto u *open source*. Operadores de redes móviles de todo el mundo participan de forma activa en diferentes consorcios que trabajan en esta línea ya que, como se ha apuntado previamente, esta visión es la que permitirá a las compañías desplegar las redes de nueva generación con mayor rapidez y reducir el tiempo de retorno de la inversión. Este es un aspecto clave, por ejemplo, en el retraso del despliegue comercial de la 5G.

Son ejemplos de estas iniciativas sin ánimo de lucro *Mosaic5G* [21] y 5G-PPP (*The 5G Infrastructure Public Private Partnership*) [22]. En ambas colaboran, junto con los operadores de telecomunicaciones, los principales proveedores tecnológicos y diversas entidades del ámbito académico como universidades e institutos de investigación.

Mosaic5G se funda con el propósito de desarrollar y promover un ecosistema de plataformas de código abierto para la implementación de soluciones RAN y de núcleo de red de cuarta y quinta generación. 5G-PPP es una iniciativa conjunta, tal y como su nombre indica, del sector privado y el sector público, representado por la Comisión Europea, que persigue establecer el liderazgo europeo en el desarrollo de la 5G y de las potenciales aplicaciones que se derivan de la nueva generación.

El movimiento hacia la 5G de estándares abiertos es también la búsqueda de la interoperabilidad de las distintas soluciones. Haciendo uso de las tecnologías de virtualización y programación de redes, los proveedores poseen los recursos para implementar los componentes de la red garantizando la compatibilidad con los creados por la competencia, de modo que los operadores no se vean limitados en la configuración de sus redes [23].

Capítulo 3

Consideraciones previas

En este capítulo se describen las líneas generales de la solución propuesta. Entre otros aspectos, se detallan los requisitos mínimos de los equipos usados para el despliegue de los núcleos de red y la configuración inicial que debe realizarse en los mismos, así como las especificaciones de la estación de trabajo utilizada en el desarrollo del proyecto. También se presentan otros dispositivos, algunos de los cuales, si bien su configuración y uso no son objeto de este Trabajo Fin de Grado, deben mencionarse pues se recurrirá a ellos para completar el prototipo de red móvil y validar el funcionamiento de los núcleos de red implementados.

3.1. Descripción general

Se pretende implementar dos EPC, núcleos de red de LTE, y un 5GC, núcleo de red de 5G, utilizando para ello las plataformas *OpenAirInterface* (OAI) y *Open5GCore*. Dado que después de su implementación los núcleos de red se integrarán con una red de acceso radio (RAN) de eNB y/o gNB, igualmente basada en *software*, para desplegar una red móvil 4G o 5G funcional, se plantean dos escenarios válidos para la interconexión de ambas partes. Estos escenarios pueden observarse en las Figuras 8 y 9.

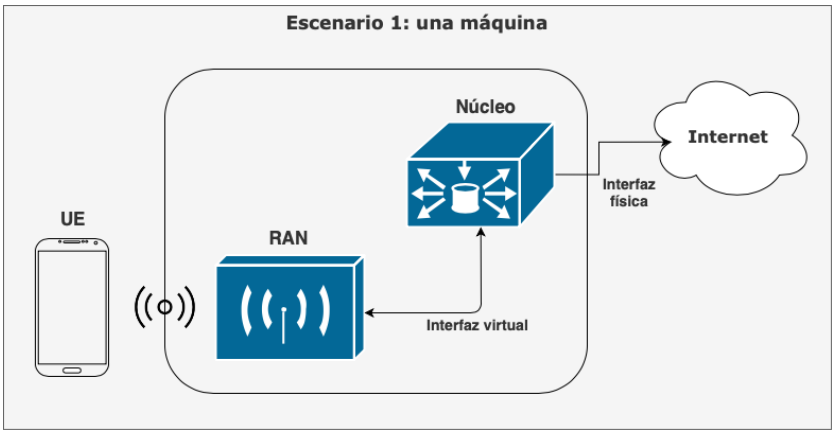

Figura 8: Arquitectura del prototipo de red móvil, despliegue en un único equipo

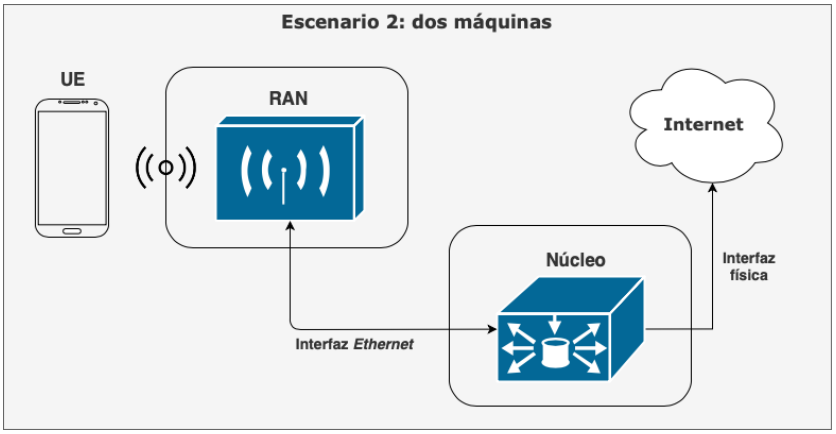

Figura 9: Arquitectura del prototipo de red móvil, despliegue en dos equipos

El primer escenario que se propone consiste en configurar el núcleo de red y la RAN en el mismo equipo, de modo que la conexión entre ambos módulos se consigue a través de interfaces de red virtuales. En cambio, la segunda propuesta contempla el uso de dos equipos, implementando en cada uno de ellos una de las partes de la red. En este caso, sí es necesaria la conexión física de las dos estaciones de trabajo; esta conexión, además, debe realizarse de forma directa a través de una interfaz *Ethernet*.

3.1.1. Requisitos mínimos

Si bien ninguna de las plataformas usadas detalla los aspectos que debe satisfacer el equipo en el que se vaya a implementar su solución de núcleo de red, en la práctica existe una serie de requisitos que el equipo en cuestión necesita reunir para completar la instalación y garantizar un funcionamiento correcto. La Tabla 1 recoge dichos requisitos.

	OpenAirInterface	**Open5GCore**
Sistema operativo	Ubuntu 16.04 / 18.04 LTS	
Procesador	2 núcleos	4 núcleos
Memoria RAM	4 GB	16 GB
Conexiones de red	Al menos una interfaz física (se recomienda Ethernet)	
Puerto USB	USB 3.0 (en despliegue conjunto con la RAN)	

Tabla 1: Requisitos mínimos para la instalación de los núcleos de red

En cuanto al sistema operativo Linux que debe ejecutar la máquina, cualquiera de las dos versiones de Ubuntu es válida. No obstante, se prefiere la más reciente 18.04 LTS, que presenta mejor compatibilidad con OAI. El sistema operativo puede ejecutarse de manera nativa en el equipo, siendo esta la opción más recomendable, pero también puede crearse una máquina virtual para el despliegue de los núcleos de red. Existen múltiples herramientas comerciales como *VMWare*, disponible para todos los sistemas operativos del mercado, o de código abierto como KVM, propia de entornos Linux.

Los recursos necesarios a nivel de procesador y memoria RAM varían en función de la implementación utilizada. De acuerdo con la documentación de OAI [24] y *Open5GCore*,

el equipo en el que se instale la solución de OAI debe disponer como mínimo de 4 GB de RAM y un procesador de dos núcleos. Estos requisitos son más restrictivos en el caso de *Open5GCore*, pues se recomiendan cuatro núcleos de procesador y 4 GB de RAM por cada uno de los módulos que componen este núcleo, ascendiendo el total de memoria RAM a 16 GB.

Para el enrutamiento del tráfico hacia redes IP externas se requiere una conexión mediante una interfaz de red física. Se recomienda hacer uso de una conexión cableada; sin embargo, es posible configurar una interfaz inalámbrica para ello. Además, como ya se ha apuntado, es necesario un puerto de red *Ethernet* adicional si se decide desplegar el escenario de la Figura 9. Por último, la estación de trabajo debe contar con un puerto USB 3.0 si se desarrolla el escenario previamente mencionado, ya que se trata de una exigencia del equipo SDR que se usa para la transmisión y recepción de señales radio.

3.1.2. Especificaciones de la estación de trabajo

El equipo sobre el que se desarrolla el proyecto es el servidor rack *SuperServer* 1029P-WTRT de la empresa *Super Micro* [25], cuyas especificaciones más reseñables en relación con los requisitos expuestos en el apartado anterior recoge la Tabla 2.

Sistema operativo	Ubuntu 18.04.4 LTS
Procesador	Intel Xeon Silver 4216 (32 núcleos @ 2,10 GHz)
Memoria RAM	64 GB
Almacenamiento	960 GB SSD
Conexiones de red	2 x 10 Gigabit Ethernet 2 x Gigabit Ethernet
Puertos USB	6 x USB 3.0

Tabla 2: Principales especificaciones del servidor de trabajo

La estación de trabajo, que tiene asignado el nombre de máquina ”mcg”, ejecuta el sistema operativo Ubuntu 18.04.4 LTS de forma nativa. La Figura 10 muestra el equipo instalado en el laboratorio del Grupo de Comunicaciones Móviles (MCG) del iTEAM.

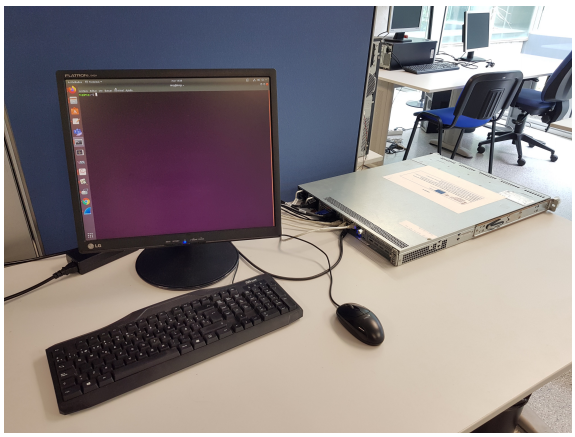

Figura 10: *SuperServer* 1029P-WTRT de *SuperMicro*

3.2. Primeros pasos

Antes de iniciar la instalación de los núcleos de red, deben completarse algunos pasos para asegurar que el sistema operativo dispone de los componentes *software* necesarios y no surgen problemas en el transcurso de la misma.

En primer lugar, es recomendable verificar que el sistema operativo cuenta con las últimas actualizaciones publicadas. Para ello, se ejecutan los siguientes comandos en una ventana de terminal de Ubuntu (`sudo` indica que son necesarios permisos de superusuario):

```
sudo apt update
sudo apt upgrade
```

Por otra parte, a lo largo del proceso de configuración de los núcleos de red será necesario administrar las interfaces de red y conexiones del sistema. En este proyecto se manejan herramientas de los paquetes `net-tools` y `iproute2`, que se pueden obtener de la siguiente forma:

```
sudo apt install net-tools iproute2
```

En relación con la gestión de las funciones de red, Linux no permite el reenvío de paquetes por defecto. Por ello, hay que configurar el *kernel* de Linux para que actúe como router y el núcleo de red implementado pueda redirigir el tráfico proveniente del terminal de usuario (UE) hacia redes externas. Este cambio se hace persistente descomentando la siguiente línea del fichero `/etc/sysctl.conf` con un editor como `nano` (`sudo nano /etc/sysctl.conf`):

```
net.ipv4.ip_forward=1
```

Los ficheros que componen la instalación tanto del EPC de OAI como de *Open5GCore* se encuentran alojados en los repositorios *GitHub* y *GitLab*, respectivamente, servicios basados en el software de control de versiones *Git*. Por tanto, se instala dicho paquete:

```
sudo apt install git
```

Finalmente, aunque no es necesario, para depurar posibles problemas en la red puede resultar conveniente capturar el tráfico de una interfaz determinada. A tal efecto se sugiere el uso de *Wireshark*.

3.3. Otros recursos *hardware*

Además del equipo de trabajo descrito previamente, que constituye el principal elemento físico utilizado para el despliegue de las soluciones propuestas, a lo largo del proyecto se emplean otros dispositivos que se presentan a continuación. Mediante ellos es posible crear el prototipo de red móvil mostrado en las Figuras 8 y 9.

Equipo SDR

La implementación de la estación base de la red móvil se lleva a cabo a través de un equipo SDR, conectado a un ordenador que cuenta con el *software* adecuado para controlar dicho equipo. En el desarrollo de este proyecto se ha hecho uso de uno de los equipos disponibles en el laboratorio del MCG, el modelo USRP B210 de la marca *Ettus Research*, que se puede observar en la Figura 11. Las características de este dispositivo pueden consultarse en la hoja de especificaciones del fabricante [26]. Aparte, se conectarán al equipo antenas que doten al mismo de conectividad LTE/5G.

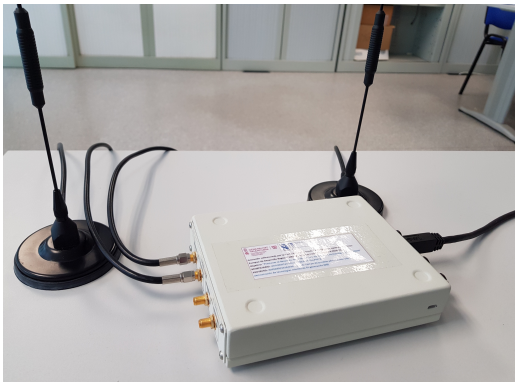

Figura 11: USRP B210 de *Ettus Research* y antenas de transmisión y recepción

Terminales móviles comerciales

Para probar el funcionamiento de los núcleos de red desplegados se emplearán dispositivos móviles comerciales como equipo de usuario (UE). En el laboratorio se dispone de *smartphones* Galaxy S10 y Galaxy A90 de Samsung. Ambos permiten la conexión a redes 4G LTE y 5G.

Tarjetas USIM

La conexión de los terminales de usuario a la red móvil desplegada requiere de tarjetas SIM para lograr la autenticación y el establecimiento de la sesión. Se utilizan tarjetas USIM (*Universal Subscriber Identity Module*) de la empresa *Open Cells* [27]. Estas tarjetas permiten configurar todos los parámetros propios de una SIM y tienen soporte para redes 2G, 3G y 4G. Pueden ser introducidas en cualquier dispositivo móvil, ya que se proporcionan con el triple formato SIM, microSIM y nanoSIM.

Open Cells también suministra un módulo USB para leer y programar las tarjetas que funciona con una herramienta de código abierto ejecutable en Linux desarrollada por la propia compañía. En la Figura 12 puede observarse una de las tarjetas USIM junto con el lector.

3.3.1. Programación de las tarjetas USIM

A continuación se resume el proceso para programar una de las tarjetas USIM empleadas en el proyecto con el programa de *Open Cells*. Cabe destacar que los parámetros que se graban en este apartado se asignarán posteriormente a un usuario. Dichos datos deberán registrarse en la entidad HSS de cada uno de los núcleos de red para que, de este

Figura 12: Tarjeta USIM y lector USB de *Open Cells*

modo, el usuario que haga uso de la tarjeta programada pueda establecer la conexión con el prototipo de red desplegado.

El primer paso es obtener la herramienta del sitio web de *Open Cells*.[1]. Tras descomprimir el fichero descargado, en una ventana de terminal de Ubuntu se navega a la ruta en la que se encuentra el mismo y se compila. En este caso la carpeta se almacena en el escritorio:

```
cd ~/Escritorio/USIM/uicc-v2.1/
make
```

El proceso es muy rápido y debería finalizar sin errores ni advertencias. Una vez completado, se procede a conectar al equipo el lector USB con la tarjeta USIM que se desea programar. En la Tabla 3 se detallan los parámetros más destacados que se pueden configurar y el valor asignado a cada uno de ellos.

El comando que debe ejecutarse para programar la tarjeta con dichos datos es el siguiente:

```
sudo ./program_uicc --adm 12345678 --imsi 208930000000003 --key
8baf473f2f8fd09487cccbd7097c6862 -xx 11111111111111111111111111111111
--spn MCG-iTEAM --authenticate
```

Parámetro	Opción del comando	Valor
ADM	–adm	12345678
IMSI	–imsi	208930000000003
OPc	-opc	8e27b6af0e692e750f32667a3b14605d
Ki	–key	8baf473f2f8fd09487cccbd7097c6862
OP	-xx	11111111111111111111111111111111
SPN	-spn	MCG-iTEAM

Tabla 3: Parámetros programables de la tarjeta USIM y valores asignados

[1] https://open-cells.com/d5138782a8739209ec5760865b1e53b0/uicc-v2.1.tgz

En el comando anterior, el código ADM (contraseña de la SIM) puede ser cualquier valor numérico de 8 dígitos, pues las tarjetas no están protegidas contra escritura. Por otra parte, solo debe añadirse como opción al ejecutar el comando uno de los parámetros (OP u OPc), ya que ambos guardan relación según la Ecuación 1, que consiste en realizar una operación criptográfica y una OR exclusiva con el parámetro OP y la clave de autenticación de la tarjeta SIM, Ki.

$$OPC = AES_{128}(k_i, OP) \oplus OP \tag{1}$$

El SPN es el nombre del proveedor de servicio y el contenido de este parámetro es el identificador de red que aparece en el dispositivo móvil. Por último, la opción `--authenticate` lanza una prueba de autenticación para obtener el SQN (*Sequence Number*). La Figura 13 muestra el mensaje de salida en la consola.

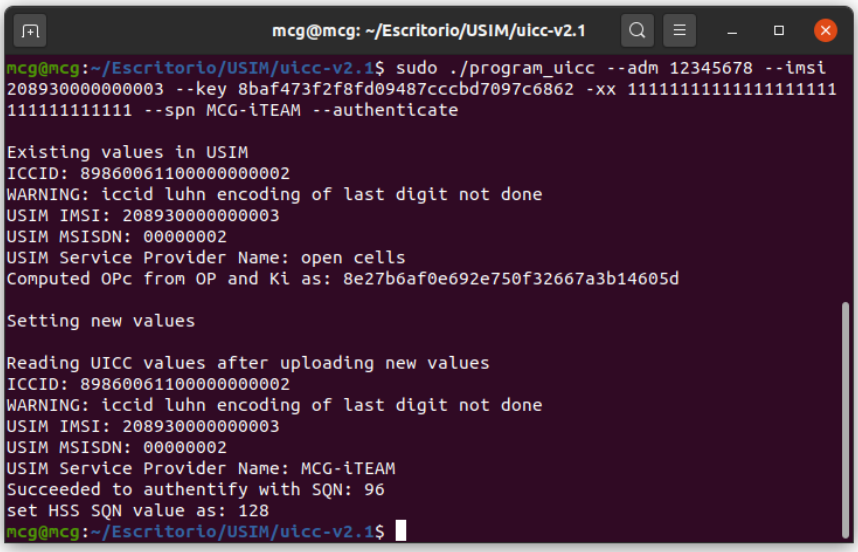

Figura 13: Ejecución de la herramienta de programación de la tarjeta USIM

De la Figura anterior puede destacarse que, tal y como se ha explicado, el valor del OPc se calcula con los datos facilitados para ejecutar el programa y coincide con el recogido en la Tabla 3. Además, se informa del SQN y de otro parámetro, el MSISDN. Ambos deben conservarse para el posterior registro del usuario en el HSS.

Capítulo 4

Despliegue del núcleo de red de *OpenAirInterface*

En este capítulo se describe el proceso de despliegue de la implementación *software* del EPC de la plataforma *OpenAirInterface* (OAI), el proyecto desarrollado por el consorcio OpenAirInterface Software Alliance (OSA). Las siguientes páginas recogen una somera revisión de las funcionalidades y las posibilidades de despliegue de la plataforma, pasando a explicar los pasos a seguir para su instalación y la configuración de la solución diseñada para el proyecto.

4.1. Distribución y funcionamiento

Los proyectos de OAI se encuentran alojados en dos servicios diferentes, aislando la implementación de la RAN del *software* del núcleo de red. El código fuente de este último, escrito principalmente en los lenguajes de programación C y C++ y desarrollado para ser ejecutado en sistemas Linux, está disponible en la página web de *GitHub* de OSA. Pueden encontrarse en el sitio web distintos repositorios. De entre todos los repositorios destacan los siguientes:

- **openair-cn:** contiene una implementación del núcleo de red LTE de acuerdo con las especificaciones de la *Release* 9 y la *Release* 10 del 3GPP, compuesta por un MME, un HSS, un P-GW y un S-GW (estos últimos ejecutados como una única instancia).
- **openair-cn-cups:** aloja la implementación CUPS (*Control and User Plane Separation*) del S-GW y el P-GW, definida por primera vez en la *Release* 14 del 3GPP, uniendo las entidades del plano de control (SGW-C y PGW-C) y del plano de usuario (SGW-U y PGW-U) en sendos componentes.
- **openair-epc-fed:** es el repositorio de referencia para el despliegue del núcleo de red de OAI a través del entorno de contenedores *Docker*, en el que se encuentran las imágenes preparadas para ello.
- **openair-k8s:** contiene recursos para el despliegue de OAI mediante el sistema de

automatización de contenedores *Kubernetes*, hallándose en fase experimental en el transcurso de la realización del presente proyecto.

El equipo de OAI trabaja para adaptar el EPC en su totalidad a las especificaciones de la *Release* 14, habiéndose desarrollado hasta este momento un nuevo HSS, que se incluye en el repositorio `openair-cn` junto al resto de elementos enumerados previamente.

Dentro de los repositorios, los recursos se organizan en torno a la siguiente estructura de directorios:

- **build:** es la carpeta en la que se compila el código fuente.
- **docs:** en este directorio pueden encontrarse diversos manuales, esquemas y líneas maestras de la implementación del núcleo de red de OAI.
- **etc:** contiene los ficheros que configuran los componentes del EPC y la base de datos de la que se sirven los primeros.
- **scripts:** los ficheros ejecutables para la instalación, configuración y ejecución de la implementación del núcleo de red se encuentran en esta carpeta.
- **src:** este directorio almacena el código fuente de la plataforma.
- **test:** contiene ficheros varios (pruebas, versiones en desarrollo, etc.).

Algunas de las características más relevantes de la propuesta de OAI para el EPC son el uso de *freeDiameter*, una implementación del protocolo de autenticación, autorización y contabilización (AAA) *Diameter*, y de certificados SSL para la comunicación entre el MME y el HSS, así como la identificación de estos dos componentes a través de FQDN (*Fully Qualified Domain Names* o nombres de dominio completo).

En adición, el HSS de OAI está desarrollado para funcionar con soporte de bases de datos. La base de datos relacional que emplea el HSS original es *MySQL*, mientras que la implementación de este componente basada en las especificaciones de la *Release* 14 requiere el uso del sistema de gestión de bases de datos NoSQL *Apache Cassandra*.

Finalmente, el funcionamiento de los distintos elementos del núcleo de red, incluida la base de datos, está regulado por ficheros de configuración en formato CONF y JSON. Entre otros aspectos, en estos archivos se definen los parámetros que identifican la red móvil y las interfaces de red y direcciones IP empleadas por los componentes para la conexión entre sí, con la RAN y con Internet.

4.2. Instalación

La instalación del EPC de OAI genera una aplicación ejecutable por componente, resultando así tres, si se despliega el núcleo con el plano de control y el plano de usuario del SPGW bajo una sola entidad, o cuatro, si se opta por la separación de estos planos.

En principio, la plataforma está diseñada para ser desplegada en un único equipo. Por ello, pese a ser posible ejecutar las aplicaciones anteriores en equipos separados siempre

y cuando estén interconectados, la instalación se lleva a cabo en la estación de trabajo utilizada a lo largo de este proyecto. Por este motivo, tampoco se considera necesario recurrir a herramientas de virtualización para aislar cada uno de los componentes del núcleo de red.

En las próximas páginas se propone la implementación del EPC basada en el principio CUPS. Sin embargo, en el transcurso del proyecto se instaló con éxito en otro equipo el *software* de OAI mediante contenedores. Con este despliegue se obtiene un núcleo de red igualmente caracterizado por la separación del plano de control y el plano de usuario.

4.2.1. Descarga del código fuente

La descarga de los ficheros de OAI puede realizarse en cualquier directorio, siempre y cuando el usuario que inicia la instalación disponga de permisos de escritura suficientes. Una opción, que es la que se desarrolla a continuación, consiste en utilizar la carpeta de usuario, cuya ruta se identifica con (`~/`).

Dado que debe descargarse código fuente de dos repositorios diferentes, se define un directorio para guardar el contenido de ambos y se usa *Git* para clonar el primero de ellos, `openair-cn`. En una ventana de terminal de Ubuntu:

```
cd ~/
sudo mkdir oai-epc
cd oai-epc/
git clone https://github.com/OPENAIRINTERFACE/openair-cn.git
```

La rama de código que se utiliza en este despliegue es la de desarrollo (`develop`). Por tanto, una vez que se dispone del código fuente de este repositorio en el equipo, debe ejecutarse nuevamente *Git* para indicarlo:

```
cd openair-cn/
git checkout develop
```

Por último, los pasos anteriores se repiten con `openair-cn-cups`, el segundo de los repositorios de los que se obtiene código:

```
cd ..
git clone https://github.com/OPENAIRINTERFACE/openair-cn-cups.git
cd openair-cn-cups/
git checkout develop
```

Tras completar estos pasos, todos los ficheros necesarios para la instalación del HSS y el MME se encuentran en la ruta `~/oai-epc/openair-cn/`, mientras que los archivos propios del SPGW-C y SPGW-U se ubican en `~/oai-epc/openair-cn-cups/`.

4.2.2. Instalación de *Cassandra*

Antes de proceder con la compilación de los elementos del EPC, se debe instalar en el equipo el *software* administrador de bases de datos *Apache Cassandra*. Cabe destacar que entre los ficheros descargados se incluye un *script* que automatiza este proceso. No obstante, las versiones de *Cassandra* disponibles actualmente requieren el entorno Java versión 8, no disponible en los repositorios de Ubuntu por razones de licencia de Oracle, la compañía propietaria. Por ello, a continuación se reproducen los comandos que hay que ejecutar para la instalación manual del citado *software*:

```
echo "deb https://downloads.apache.org/cassandra/debian 311x main" |
sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
curl https://downloads.apache.org/cassandra/KEYS | sudo apt-key add -
sudo apt update
sudo apt install cassandra
```

Durante el proceso se instalan las dependencias (paquetes necesarios para el correcto funcionamiento de la aplicación), entre ellas una versión de código abierto del entorno Java. Para verificar que la instalación se ha completado satisfactoriamente, puede lanzarse el comando `nodetool status`, tras lo que debería aparecer en la ventana de terminal una respuesta como la de la Figura 14.

```
mcg@mcg:~$ nodetool status
Datacenter: DC1
===============
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Tokens       Owns (effective)  Host ID                               Rack
UN  127.0.0.1  279,92 KiB  256          100,0%            61027481-ddd7-4c79-93f4-920c98e3de26  rack1
```

Figura 14: Servidor de *Cassandra* ejecutándose en el equipo

4.2.3. Compilación de los componentes del EPC

El último paso en el proceso de instalación de la plataforma OAI es la compilación de las entidades que forman la implementación del EPC. Los archivos ejecutables que inician la compilación de los componentes anteriores se encuentran en el directorio `scripts/` de cada uno de los repositorios descargados.

Es necesario ejecutar estos *scripts* dos veces: la primera, para instalar dependencias requeridas por los elementos del núcleo de red, mediante las opciones `-i` y `f`, y la segunda, para dar comienzo a la compilación (opción `-c`).

Además, en la compilación del SPGW-C y el SPGW-U es necesario señalar con la opción `-b` la versión que se desea compilar, debiendo indicar `Release` en ambos casos. En resumen, los comandos que deben ejecutarse son los siguientes:

```
cd ~/oai-epc/openair-cn/scripts/
sudo ./build_hss_rel14 -i -f
sudo ./build_hss_rel14 -c
sudo ./build_mme -i -f
sudo ./build_mme -c
cd ~/oai-epc/openair-cn-cups/build/scripts
sudo ./build_spgwc -I -f
sudo ./build_spgwc -c -V -b Release -j
sudo ./build_spgwu -I -f
sudo ./build_spgwu -c -V -b Release -j
```

Si el proceso transcurre sin problemas, la ventana de terminal muestra un mensaje como el de la Figura 15 tras la compilación de cada uno de los componentes.

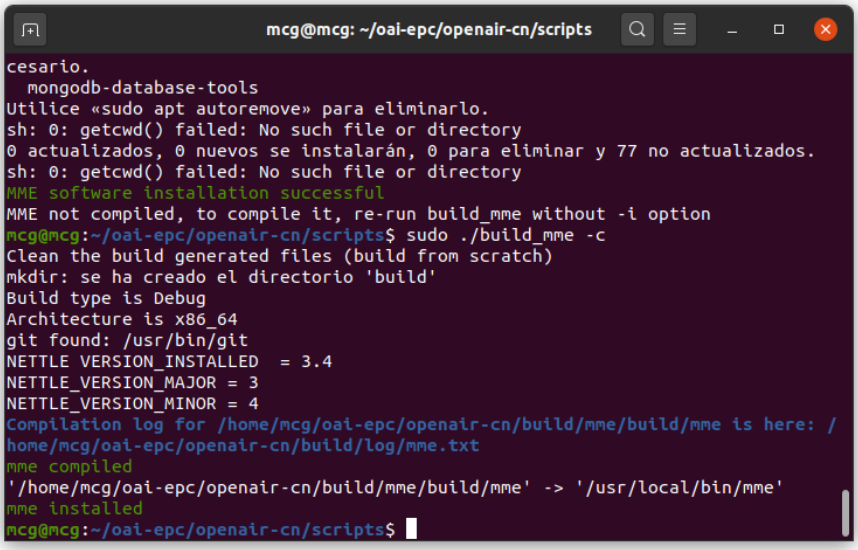

Figura 15: Compilación del MME finalizada

4.3. Configuración

Completada la instalación del núcleo de red, el siguiente aspecto a tratar es su configuración. Ello implica modificar los archivos de configuración ubicados en el directorio `etc/` de la instalación y adaptar la configuración de red del equipo a lo indicado en estos. También es necesario configurar *Cassandra* para trabajar con el *software* de OAI y crear la base de datos que utiliza el HSS, en las que se registran los usuarios para permitir su conexión al prototipo de red móvil.

4.3.1. Propuesta de configuración de red del EPC

Con el objetivo de facilitar al lector la comprensión de la configuración aplicada al núcleo de red desplegado, se presenta el diagrama de la Figura 16. En el mismo pueden observarse los componentes implementados en el EPC de OAI y las direcciones IP asignadas a cada una de las interfaces de red implementadas.

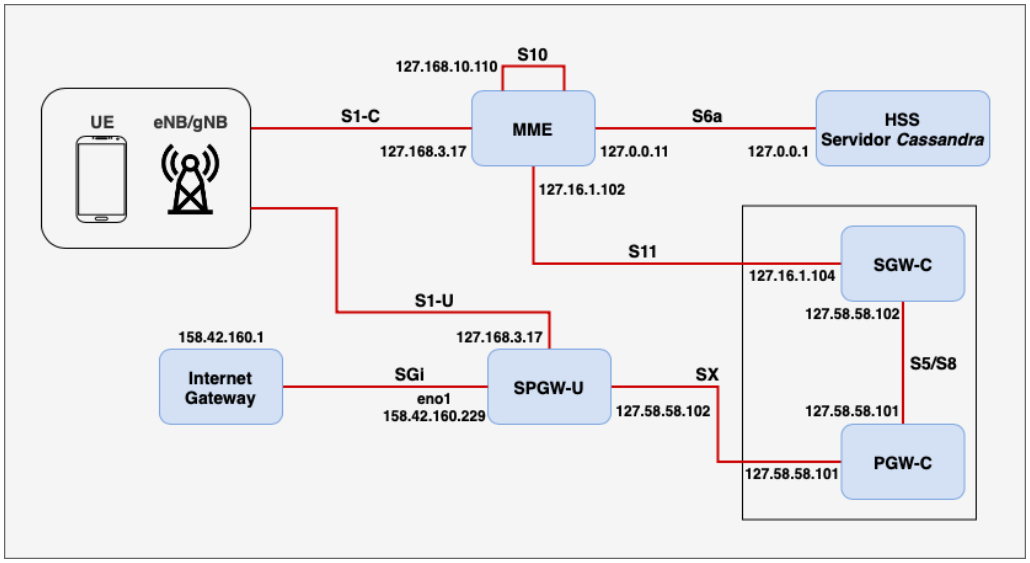

Figura 16: Descripción general de la implementación del EPC de OAI

Del diagrama anterior cabe destacar el uso de la interfaz virtual *loopback* (`lo`). Su dirección de red es `127.0.0.0/8` y las direcciones IP de este rango se utilizan cuando el tráfico generado por una aplicación tiene como destino la propia máquina. Dado que el despliegue se realiza sobre un único equipo, no es necesario configurar interfaces adicionales y pueden asignarse direcciones de este rango para la comunicación entre los componentes del núcleo de red.

En la Figura 16 también se aprecia la identificación del HSS con el servidor de *Cassandra* que, como el resto de componentes, se despliega en el mismo equipo. Así pues, la dirección IP de la interfaz S6A es la dirección IP del servidor de bases de datos, `127.0.0.1`.

Por otro lado, la implementación de OAI de la interfaz SGi del SPGW-U definida para el reenvío hacia redes IP externas del tráfico generado por los usuarios debe corresponderse con una interfaz física del equipo, existiendo la posibilidad de elegir entre una interfaz inalámbrica o un puerto *Ethernet*. En este caso, se asigna la interfaz `eno1` de la estación de trabajo, a través de la cual la máquina establece conexión con la red de la UPV e Internet. La dirección IP del *gateway* o puerta de enlace es `158.42.160.1`.

Finalmente, la configuración que se propone está pensada para el escenario de la Figura 8, en el que la RAN se halla en el mismo equipo que el EPC. En caso de querer desplegar la solución de la Figura 9, con la RAN en otra máquina, las interfaces S1-C del MME y S1-U del SPGW-U deben identificarse con una interfaz *Ethernet* de la estación de trabajo, asignando convenientemente una dirección IP a la misma para la conexión con el otro equipo.

4.3.2. Configuración de *Cassandra*

Tras la instalación de *Cassandra*, debe configurarse el servidor para su integración con la plataforma de OAI. Para ello, en primer lugar se detiene el servicio y se eliminan algunos archivos existentes. En una ventana de terminal de Ubuntu:

```
sudo service cassandra stop
sudo rm -r /var/log/cassandra/*
sudo rm -r /var/lib/cassandra/*
```

A continuación, se procede a editar el fichero de configuración en formato YAML de *Cassandra*, ubicado en `/etc/cassandra/` con un editor como `nano`:

```
cd /etc/cassandra/
sudo nano cassandra.yaml
```

Deben editarse las siguientes líneas (los puntos suspensivos indican contenido en el fichero entre las líneas a las que se hace referencia):

```
...
cluster_name: "HSS Cluster"
...
listen_address: 127.0.0.1
...
start_rpc: true
rpc_address: 0.0.0.0
broadcast_rpc_address: 127.0.0.1
...
endpoint_snitch: GossipingPropertyFileSnitch
```

Completada la edición, se vuelve a iniciar el servicio de *Cassandra* y se verifica su estado:

```
sudo service cassandra start
sudo service cassandra status
```

El estado que debe aparecer como respuesta al comando es `active (running)`. Esta comprobación, junto con el uso de `nodetool status`, al que debe acompañar una respuesta como la de la Figura 14, sirve para descartar posibles problemas en la ejecución del servidor de *Cassandra*. En caso contrario, no podrán registrarse usuarios en la base de datos ni iniciará el proceso del HSS.

Si surgen problemas con *Cassandra*, la forma más efectiva de abordarlos es detener el servicio y ejecutar de forma directa la aplicación (`cassandra` en la ventana de terminal) para identificar los errores.

4.3.3. Configuración del HSS

Las aplicaciones de OAI están programadas para obtener los ficheros de configuración en una ruta específica, `/usr/local/etc/oai/`. Por ello, antes de proceder con la configuración de los distintos componentes debe crearse dicho directorio, dotando al usuario de permisos suficientes para su modificación:

```
Ruta='/usr/local/etc/oai'
sudo mkdir -p $Ruta
sudo mkdir $Ruta/freeDiameter
sudo chmod -R 777 $Ruta
```

Una vez creado el directorio se copian en el mismo las plantillas de los archivos CONF y JSON disponibles en la carpeta `etc` de la instalación. Los ficheros que configuran el uso que hacen el HSS y el MME del protocolo *freeDiameter* se ubican en una carpeta separada. Los comandos a ejecutar son los siguientes:

```
cd ~/oai-epc/openair-cn/etc/
sudo cp acl.conf hss_rel14_fd.conf $Ruta/freeDiameter
sudo cp hss_rel14.conf hss_rel14.json oss.json $Ruta
```

Por tanto, son cinco los archivos de configuración del HSS. Los parámetros más importantes, en cuya configuración debe intervenir el usuario, están identificados mediante el símbolo arroba (`@`) en estos ficheros y el resto de los que definen el funcionamiento de los componentes del EPC de OAI. Salvo que se busquen opciones avanzadas, el resto de variables no requieren modificación por parte del usuario.

En el caso del HSS, estos parámetros son los siguientes y se les asignan los valores recogidos en la Tabla 4:

- `PREFIX`: es la ruta en la que se encuentran los ficheros de configuración y certificados.
- `REALM`: especifica el dominio del FQDN del HSS.
- `HSS_FQDN`: es el FQDN del HSS.
- `cassandra_Server_IP`: indica la dirección IP del servidor de *Cassandra*.
- `OP_KEY`: se informa del código OP que usa el HSS para la autenticación, por lo que la tarjeta USIM se programa con el mismo valor (Tabla 3).
- `ROAMING_ALLOWED`: indica si el núcleo de red admite la función de itinerancia.

Adicionalmente, se realizan los siguientes cambios:

- Se descomenta la línea 70 del fichero `hss_rel14_fd.conf` (`ListenOn = 127.0.0.1`).
- En el archivo `hss_rel14.json` se cambia el valor de la variable `reloadkey` a `true`.

- En el fichero anterior se modifica la ruta de las variables `logname` (tres en total), de modo que la nueva ruta de los archivos LOG es `/var/log/`.
- También se indica en la variable `ossfile` del archivo `hss_rel14.json` la ruta en la que efectivamente se encuentra dicho fichero, `/usr/local/etc/oai/oss.json`.

Variable	Valor
PREFIX	/usr/local/etc/oai
REALM	openairinterface.org
HSS_FQDN	hss.openairinterface.org
cassandra_Server_IP	127.0.0.1
OP_KEY	11111111111111111111111111111111
ROAMING_ALLOWED	true

Tabla 4: Parámetros configurados en el HSS

Los ficheros de registro del HSS deben crearse en la ruta especificada previamente, para lo que se lanzan los siguientes comandos:

```
sudo touch /var/log/hss.log
sudo touch /var/log/hss_stat.log
sudo touch /var/log/hss_audit.log
```

Finalmente, se instalan los certificados SSL del HSS requeridos para la conexión con el MME por medio de *freeDiameter*:

```
cd ~/oai-epc/openair-cn/src/hss_rel14/bin/
sudo ./make-certs.sh hss openairinterface.org $Ruta
```

4.3.4. Registro de usuarios en la base de datos del HSS

Añadir usuarios a la base de datos del HSS para permitir su conexión a la red móvil desplegada es un proceso automatizado por un par de *scripts* programados por el equipo de desarrollo de OAI. En primer lugar, debe crearse la base de datos en el servidor de *Cassandra* instalado. Para ello, únicamente es necesario cargar el fichero que contiene las tablas a implementar en la base de datos:

```
cqlsh --file ~/oai-epc/openair-cn/src/hss_rel14/db/oai_db.cql 127.0.0.1
```

Una vez importadas las tablas, se ejecutan los *scripts* disponibles en el directorio del mismo nombre en la carpeta de instalación. Este es el comando que debe ejecutarse para añadir a la base de datos al usuario cuya tarjeta USIM tiene programados los datos de la Tabla 3:

```
Cassandra_Server_IP='127.0.0.1'
Dominio='openairinterface.org'
cd ~/oai-epc/openair-cn/scripts/
./data_provisioning_users --apn oai.ipv4 --apn2 internet
--key 8baf473f2f8fd09487cccbd7097c6862 --imsi-first 208930000000003
--msisdn-first 000002 --mme-identity mme.$Dominio --no-of-users 1
--realm $Dominio --truncate False  --verbose True
--cassandra-cluster $Cassandra_Server_IP
```

En el comando anterior deben indicarse, entre otros, un APN (*Access Point Name*) primario y secundario asociados al usuario, el MSISDN que devuelve la herramienta de programación de las tarjetas USIM (Figura 13) y el número de usuarios que se desea registrar, añadiéndose con IMSI y MSISDN consecutivos a los indicados. La opción `Truncate` configurada a `False` garantiza que no se borran los datos previamente existentes en la base de datos.

Por último, se lanza el siguiente *script* para registrar la identidad del MME al que se conecta el HSS:

```
./data_provisioning_mme --id 3 --mme-identity mme.$Dominio
--realm $Dominio --ue-reachability 1 --truncate True
--verbose True -C $Cassandra_Server_IP
```

Si el proceso se completa correctamente, en la ventana de terminal aparecen los mensajes de la Figura 17. En caso contrario, es posible que el servidor de *Cassandra* no esté ejecutándose correctamente, para lo que se recomienda revisar el apartado destinado a la aplicación.

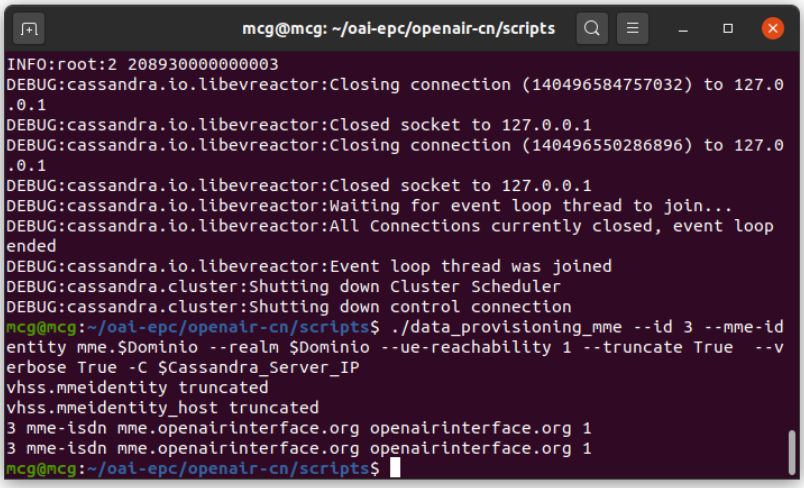

Figura 17: Usuario registrado en la base de datos del HSS de OAI

4.3.5. Configuración del MME

La configuración del MME se establece en dos ficheros CONF, los cuales deben copiarse en la ruta definida por OAI para los mismos:

```
Ruta='/usr/local/etc/oai'
sudo cp acl.conf mme_fd_sprint.conf $Ruta/freeDiameter/mme_fd.conf
sudo cp mme.conf $Ruta
```

Los parámetros generales del MME que deben definirse son los siguientes:

- `INSTANCE`: especifica la instancia de MME que se configura.
- `PREFIX`: es la ruta en la que se encuentran los ficheros de configuración y certificados.
- `REALM`: especifica el dominio del FQDN del MME.
- `PID_DIRECTORY`: es la ruta del directorio de trabajo de un proceso en Linux.
- `MME_FQDN`: es el FQDN del MME.
- `HSS_HOSTNAME`: especifica el nombre de máquina del FQDN del HSS.
- `HSS_FQDN`: es el FQDN del HSS.
- `HSS_IP_ADDR`: indica la dirección IP del HSS, que se corresponde con la del servidor de *Cassandra*.
- `OUTPUT`: especifica la forma de visualización del registro de actividad (en la ventana de terminal o en un archivo).

Los valores asignados a las variables anteriores en la configuración propuesta aparecen en la Tabla 5. Asimismo, existen otros parámetros que identifican de forma unívoca al MME de una red móvil y deben coincidir con los códigos registrados en la estación base para poder conectar la RAN con el EPC y conseguir un despliegue efectivo de la red. Estos códigos son:

- `MCC`: especifica el *Mobile Country Code* o código de identificación de país del operador de red.
- `MNC`: especifica el *Mobile Network Code* o código de identificación de red del operador.
- `MME_GID`: es el código de identidad del grupo del MME.
- `MME_CODE`: es el código del MME dentro del grupo.
- `TAC_0`: especifica el *Tracking Area Code* o área de seguimiento dentro de la red móvil a la que da servicio el MME.
- `TAC_1` y `TAC_2`: especifica otros TAC gestionados por el MME.

Variable	Valor
INSTANCE	1
PREFIX	/usr/local/etc/oai
REALM	openairinterface.org
PID_DIRECTORY	/var/run
MME_FQDN	mme.openairinterface.org
HSS_HOSTNAME	hss
HSS_FQDN	hss.openairinterface.org
HSS_IP_ADDR	127.0.0.1
OUTPUT	CONSOLE

Tabla 5: Parámetros de ejecución configurados en el MME

Variable	Valor
MCC	208
MNC	93
MME_GID	32768
MME_CODE	3
TAC_0	600
TAC_1	601
TAC_2	602

Tabla 6: Parámetros de identificación de red configurados en el MME

Los códigos asignados a los parámetros anteriores pueden encontrarse en la Tabla 6. Se asigna el MCC y MNC codificados en el IMSI de la tarjeta USIM programada, de modo que el UE pueda conectarse al prototipo de red móvil.

Resta configurar las diferentes interfaces del equipo utilizadas para las interfaces definidas por el estándar LTE para la conexión con el resto de entidades del EPC y con los eNB y/o gNB. En el fichero `mme_fd.conf` debe especificarse la dirección IP que se asigna al MME para la interfaz S6A, mientras que en el otro archivo, `mme.conf`, se configura el resto de interfaces que conectan el MME. La configuración de red propuesta, que se identifica con la de la Figura 16, está resumida en la Tabla 7.

Destaca especialmente la interfaz física y la dirección IP asignadas para la interfaz S1-C que conecta el MME con las estaciones base, puesto que es la dirección IP de referencia para que el eNB o gNB establezca conexión con el núcleo de red. Asimismo, hay que indicar la dirección IP de la interfaz S11 del SGW-C.

Interfaz	Parámetro	Valor
S6A	MME_S6A_IP_ADDR	127.0.0.11
S1-C	MME_INTERFACE_NAME_FOR_S1_MME	lo
	MME_IPV4_ADDRESS_FOR_S1_MME	127.168.3.17/8
S11 (MME)	MME_INTERFACE_NAME_FOR_S11	lo
	MME_IPV4_ADDRESS_FOR_S11	127.16.1.102/8
S11 (SGW-C)	SGW_IPV4_ADDRESS_FOR_S11_TEST_0	127.16.1.104/8
	SGW_IPV4_ADDRESS_FOR_S11_0	
S10	MME_INTERFACE_NAME_FOR_S10	lo
	MME_IPV4_ADDRESS_FOR_S10	127.168.10.110/8
	PEER_MME_IPV4_ADDRESS_FOR_S10_0	0.0.0.0/24
	PEER_MME_IPV4_ADDRESS_FOR_S10_1	

Tabla 7: Configuración de las interfaces de red del MME

Además de los cambios descritos hasta ahora, debe modificarse en el fichero `mme.conf` el apartado final, `WRR_LIST_SELECTION`. El contenido de esta sección tras los cambios es el siguiente (los puntos suspensivos indican contenido omitido en la línea):

```
{ID="tac-lb07.tac-hb00.tac.epc.mnc001.mcc001.3gppnetwork.org" ...
{ID="tac-lb58.tac-hb02.tac.epc.mnc093.mcc208.3gppnetwork.org" ...
{ID="tac-lb59.tac-hb02.tac.epc.mnc093.mcc208.3gppnetwork.org" ...
{ID="tac-lb5a.tac-hb02.tac.epc.mnc093.mcc208.3gppnetwork.org" ...
```

La puesta a punto del MME se completa generando los certificados SSL necesarios para la comunicación mediante el protocolo *freeDiameter* con el HSS. Para ello, se lanza el siguiente *script* (la ruta y el FQDN del MME deben especificarse al ejecutar el comando):

```
cd ~/oai-epc/openair-cn/scripts/
sudo ./check_mme_s6a_certificate
$Ruta/freeDiameter mme.openairinterface.org
```

4.3.6. Configuración del SPGW-C

Los parámetros del SPGW-C están definidos en un único fichero CONF, que debe estar ubicado junto al resto de archivos de configuración del EPC de OAI. Por tanto, se copia la plantilla que se encuentra en la carpeta `etc` de la instalación de `openair-cn-cups`:

```
cd ~/oai-epc/openair-cn-cups/etc/
sudo cp spgw_c.conf /usr/local/etc/oai/
```

Exceptuando los parámetros `INSTANCE` y `PID_DIRECTORY`, explicados en el desarrollo de la configuración del MME y que toman en este caso los mismos valores (véase la Tabla 5), el resto de variables están relacionadas con las interfaces de red implementadas en el SPGW-C. Son cuatro en total, siguiendo las especificaciones del estándar LTE para el

EPC, y la propuesta de configuración se presenta en la Tabla 8.

Interfaz	Variable	Valor
S11	SGW_INTERFACE_NAME_FOR_S11	lo
	IPV4_ADDRESS	127.16.1.104/8
S5/S8 (SGW-C)	SGW_INTERFACE_NAME_FOR_S5_S8	lo
	IPV4_ADDRESS	127.58.58.102/8
S5/S8 (PGW-C)	PGW_INTERFACE_NAME_FOR_S5_S8	lo
	IPV4_ADDRESS	127.58.58.101/8
SX	PGW_INTERFACE_NAME_FOR_SX	lo
	IPV4_ADDRESS	127.55.55.101/8

Tabla 8: Configuración de las interfaces de red del SPGW-C

Otro aspecto que es necesario configurar en el SPGW-C para que el núcleo de red funcione correctamente es el servidor DNS (sistema de nombres de dominio) utilizado para resolver los nombres de dominio requeridos por el usuario que navega por Internet. La dirección o direcciones IP que se incluyan en el fichero tienen prioridad respecto a la configuración DNS del equipo que aloja el despliegue.

OAI requiere la dirección de dos servidores, uno primario y otro secundario. Puede indicarse la dirección IP de la red a la que está conectada la estación de trabajo, o la del servidor DNS de Google (`8.8.8.8`). En la implementación llevada a cabo, se recurre a los servidores DNS de la UPV:

```
DEFAULT_DNS_IPV4_ADDRESS     = "158.42.248.88";
DEFAULT_DNS_SEC_IPV4_ADDRESS = "158.42.1.8";
```

Los rangos de direcciones IP reservadas para los UE que se autentifiquen en la red también pueden modificarse, aunque se ha optado por dejar la configuración por defecto. Por último, debe configurarse la lista de APN registrados para que coincidan con los APN asociados al usuario registrado. De este modo, el apartado `APN_LIST` queda tras la edición como sigue (los puntos suspensivos indican contenido omitido en la línea):

```
{APN_NI = "oai.ipv4" ...
{APN_NI = "internet" ...
```

4.3.7. Configuración del SPGW-U

La configuración del SPGW-U es similar a la del SPGW-C. En primer lugar y, como con el resto de componentes de la implementación del EPC de OAI, debe copiarse el correspondiente fichero de configuración en la ruta definida por el *software*:

```
cd ~/oai-epc/openair-cn-cups/etc/
sudo cp spgw_u.conf /usr/local/etc/oai/
```

Nuevamente, los parámetros `INSTANCE` y `PID_DIRECTORY` mantienen los valores de componentes anteriores, mientras que las tres interfaces de LTE implementadas por OAI en el SPGW-U se configuran de acuerdo con lo recogido en la Tabla 9.

Interfaz	Variable	Valor
S1-U	SGW_INTERFACE_NAME_FOR_S1U_S12_S4	lo
	IPV4_ADDRESS	127.168.3.17/8
SX	SGW_INTERFACE_NAME_FOR_SX	lo
	IPV4_ADDRESS	172.55.55.102/8
SGi	SGW_INTERFACE_NAME_FOR_SGI	eno1

Tabla 9: Configuración de las interfaces de red del SPGW-U

Conviene comentar algunos aspectos de la Tabla anterior. Por un lado, puede observarse que la dirección IP asignada al SPGW-U en la interfaz S1-U coincide con la fijada para el MME en la interfaz S1-C (Tabla 7). No existe inconveniente alguno en ello, dado que la implementación de OAI utiliza puertos de red diferentes para cada una de las comunicaciones. Este, además, es el esquema que se seguiría en caso de disponer de una RAN ajena al equipo, dado que la conexión se efectuaría a través de una interfaz *Ethernet* a la que se le asignaría una única dirección IP.

Por otra parte, la interfaz de red que se usa para la interfaz SGi del SPGW-U debe ser aquella que tenga una puerta de enlace hacia la red IP externa. Como ya se ha indicado al principio de esta sección, `eno1` es el puerto *Ethernet* mediante el que la estación de trabajo establece conexión con la red de la UPV y, en última instancia, con Internet, por lo que es el escogido.

Para garantizar que el tráfico generado por los usuarios es redireccionado correctamente, es necesario que el equipo realice el enmascaramiento de las direcciones IP asignadas por el SPGW-C a los UE. Este mecanismo se denomina NAT (*Network Address Translation*) y se activa por separado para cada interfaz de red del equipo añadiendo la entrada correspondiente a las tablas IP mediante el siguiente comando:

```
sudo iptables -t nat -A POSTROUTING -s 158.42.160.229/24
-o eno1 -j MASQUERADE
```

Aunque en principio este paso no es necesario si las tablas de enrutamiento del equipo están bien definidas, ante problemas de conexión del UE puede definirse una tabla de enrutamiento cuya única entrada sea reenviar el tráfico con dirección a la puerta de enlace de la Figura 16. Junto a ello, se añade una regla para que el tráfico procedente del UE siga las instrucciones de dicha tabla.

```
echo '200 lte' | sudo tee --append /etc/iproute2/rt_tables
sudo ip r add default via 158.42.160.1 dev eno1 table lte
sudo ip rule add from 12.0.0.0/8 table lte
```

Finalmente, la configuración del SPGW-U concluye editando `PDN_NETWORK_LIST` para que coincida con los rangos de direcciones disponibles para los UE indicados en el fichero de

configuración del SPGW-C. Como en el anterior no se ha modificado nada, no es necesario realizar cambios. En la última línea del fichero `spgw_u.conf` aparece el parámetro para especificar la dirección IP de la interfaz SX del SPGW-C que conecta el plano de control y de usuario de estas entidades. Se toma de la Tabla 8.

4.4. Ejecución

Terminada la configuración del núcleo de red, los distintos elementos de la implementación están preparados para ser ejecutados. Durante la compilación de los mismos se crean enlaces a las aplicaciones en el directorio `/bin/` del sistema, que se usan ahora para lanzar los procesos.

Cada proceso ha de ejecutarse en una ventana de terminal de Ubuntu, especificando la ruta del archivo de configuración del componente en cuestión. Es de suma importancia lanzar los componentes de acuerdo con la siguiente secuencia:

1. **HSS**

```
sudo oai_hss -j /usr/local/etc/oai/hss_rel14.json
```

2. **MME**

```
cd ~/oai-epc/openair-cn/scripts
sudo ./run_mme -c /usr/local/etc/oai/mme.conf -s
```

3. **SPGW-C**

```
sudo spgwc -o -c /usr/local/etc/oai/spgw_c.conf
```

4. **SPGW-U**

```
sudo spgwu -o -c /usr/local/etc/oai/spgw_u.conf
```

En este punto, el núcleo de red LTE de OAI estará operativo con la configuración definida por el usuario. El registro de actividad de cada uno de los componentes se puede consultar en la respectiva ventana de terminal. Para detener la ejecución de los procesos, se utiliza el atajo de teclado `Ctrl+C` como con cualquier otro proceso lanzado en dicho entorno.

Capítulo 5

Despliegue de *Open5GCore*

Este capítulo aborda la instalación y puesta a punto de *Open5GCore*, la implementación *software* de núcleos de red LTE y 5G del instituto alemán Fraunhofer FOKUS. A lo largo de las próximas páginas se detalla el proceso seguido para su despliegue y se presentan los aspectos más susceptibles de ser configurados por su relevancia en el funcionamiento del sistema.

5.1. Distribución y funcionamiento

El código fuente de la plataforma, mayoritariamente escrito en el lenguaje de programación C para su ejecución en entornos Linux, está disponible en el repositorio de Fraunhofer FOKUS en *GitLab*. Los ficheros de la distribución están repartidos en tres directorios principales:

- **open5gcoreRel3:** contiene los archivos de configuración, bases de datos y *scripts* utilizados para implementar el EPC.
- **open5gcoreRel5:** en este directorio se encuentran los ficheros y bases de datos que configuran el 5GC.
- **phoenix:** contiene todos los recursos de la plataforma del mismo nombre sobre la que se ejecutan los distintos componentes de los núcleos de red anteriores, además de réplicas de los archivos disponibles en las carpetas anteriores.

En resumen, el funcionamiento de las dos versiones de núcleo de red disponibles en *Open5GCore* está completamente basado, por tanto, en el entorno *phoenix* y sus módulos desarrollados por el equipo de FOKUS. Cada componente del EPC (entidad) o del 5GC (función de red) se corresponde con un proceso de dicha plataforma. Cada uno de estos procesos se diferencia en su comportamiento del resto ya que ejecutan unos módulos u otros dependiendo de los archivos de configuración de *phoenix* que definen los distintos componentes.

En el caso de la *Release* 3, dichos ficheros están escritos en formato XML, mientras que los de la *Release* 5 tienen formato JSON. Adicionalmente, *Open5GCore* hace uso de

diferentes bases de datos gestionadas a través de *MySQL*, bien para almacenar determinados parámetros y que puedan emplearse en los ficheros de configuración descritos o bien para completar la funcionalidad de algún componente (por ejemplo, el HSS por su propia definición de base de datos de usuarios).

La estructura del código fuente en el repositorio también guarda relación con las distintas modalidades que se ofrecen para realizar la instalación del *software*. Es posible hacer el despliegue en varios equipos interconectados, de modo que cada uno actúe como un componente del núcleo de red, o en una única máquina. Si se opta por la segunda opción, existen a su vez múltiples posibilidades, radicando las diferencias entre las mismas en la herramienta de virtualización empleada para simular cada elemento:

- **KVM:** mediante archivos de imagen de las distintas máquinas virtuales basadas en *kernel*, totalmente configuradas, disponibles para descarga en ambas versiones.
- **OpenStack:** la *Release* 5 puede desplegarse por medio de esta plataforma de computación en la nube a partir del paquete proporcionado por FOKUS.
- **Docker y Kubernetes:** tanto la *Release* 3 como la *Release* 5 pueden ejecutarse en entornos de contenedores haciendo uso de las imágenes preparadas para ser descargadas.
- **chroot:** pese a no ser una herramienta de virtualización propiamente dicha, a través de esta operación disponible en sistemas UNIX se crea una copia del sistema, denominada jaula, en un directorio del mismo, de modo que aquellos procesos que se ejecutan en su interior quedan aislados.

Así pues, los paquetes preparados para instalar *Open5GCore* a través de alguna de las opciones anteriores contienen el directorio correspondiente a la versión escogida junto con los módulos de *phoenix*, mientras que la instalación física en varios equipos o en una única máquina con *chroot* requiere la descarga de todo el código fuente disponible en el repositorio.

Asimismo, las subredes que conectan los distintos componentes que integran los núcleos de red están definidas de antemano. En el supuesto de llevar a cabo el despliegue mediante contenedores o recurriendo a las posibilidades que ofrece *chroot*, esta arquitectura de componentes *software* interconectados y compartiendo recursos *hardware* se construye gracias a una característica presente en el *kernel* Linux, los espacios de nombres de red o *network namespaces*. Esta función resulta esencial y con ella se generan espacios aislados para la ejecución de procesos y la configuración de red.

5.2. Instalación

Puesto que para la realización del proyecto se ha decidido utilizar una única estación de trabajo, se descarta la instalación de *Open5GCore* en varios equipos. Por otro lado, se ha preferido no utilizar herramientas de virtualización en la solución final, de modo que la forma de instalación que se expone consiste en crear una jaula *chroot* y ejecutar dentro de ella las distintas partes del núcleo de red en sus respectivos espacios de nombres de red.

Optar por esta forma de despliegue sobre una única máquina es factible porque el servidor del laboratorio tiene recursos suficientes para soportar la carga computacional. Además, renunciar a las máquinas virtuales permite, en teoría, aprovechar mejor dichos recursos y, sobre todo, otorga la capacidad de realizar una única instalación para disponer de ambos núcleos de red (LTE y 5G). En cambio, las otras posibilidades obligan a elegir una de las dos implementaciones. No obstante, durante el desarrollo del proyecto se llevó a cabo una instalación funcional de la *Release* 3 sobre KVM.

5.2.1. Descarga del código fuente

Los ficheros de *Open5GCore* se instalan en el directorio `/opt/`, siendo necesario establecer como propietario de dicha carpeta y subcarpetas al usuario que inicia la instalación (en este caso, el usuario del equipo es "mcg"). Para ello, se ejecuta en una ventana de terminal de Ubuntu el comando `chown` con la opción `-R`. Tras preparar el directorio, se usa *Git* para clonar el repositorio y disponer del código fuente en el equipo:

```
cd /opt/
sudo chown -R mcg /opt/
git clone https://gitlab.fokus.fraunhofer.de/phoenix/customer/<name>.git
```

En el comando anterior, `<name>` se sustituye por la etiqueta del cliente que ha adquirido la licencia del *software*. A continuación, debe crearse un enlace simbólico al directorio `phoenix` recién descargado en la carpeta `/opt/` puesto que el código está preparado para ser lanzado desde dicha ruta:

```
ln -s <name>/phoenix/ phoenix
```

Por tanto, a partir de este momento todos los comandos se ejecutan en rutas que cuelgan del enlace simbólico `/opt/phoenix/`.

5.2.2. Compilación de *phoenix*

Tras completar la descarga del código en el directorio correspondiente, el siguiente paso es compilar *phoenix* para poder usar la plataforma. Antes de proceder con ello, hay que lanzar el *script* `prereq.sh`. Este ejecutable se encarga de instalar las dependencias de *Open5GCore*.

```
cd phoenix/
sudo ./prereq.sh
```

El proceso termina satisfactoriamente después de ejecutar dos veces el *script* anterior, tras lo que debe aparecer un mensaje anunciando la creación de un directorio `build`, en el que se compila el código. Para evitar posibles fallos durante la compilación o en la ejecución posterior, es necesario dotar a la carpeta de permisos de ejecución mediante el siguiente comando:

```
sudo chmod 777 build
```

La compilación se inicia desde la carpeta `build` a través de estos comandos, indicando los puntos (`..`) que el código fuente se encuentra en el directorio inmediatamente superior, esto es, `/opt/phoenix/`.

```
cd build/
cmake ..
make -j`nproc`
```

El proceso se da por completado cuando la ventana de terminal muestra un mensaje como el de la Figura 18.

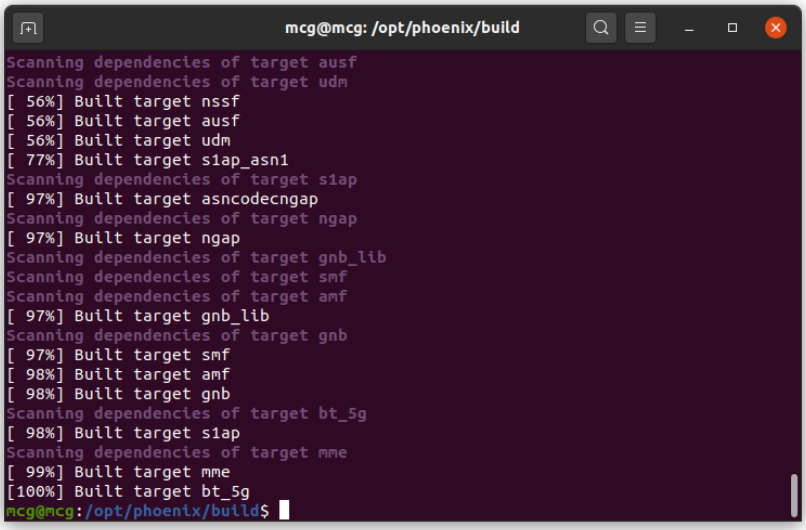

Figura 18: Compilación de *phoenix* finalizada

5.2.3. Creación de la jaula *chroot*

El entorno en el que se ejecutarán las diferentes instancias de *phoenix* se encuentra comprimido en `/opt/phoenix/tools/ph_init/chroot/` y puede extraerse haciendo uso del *script* preparado para ello en la ubicación anterior:

```
cd ../tools/ph_init/chroot/
sudo ./prepare-once.sh
```

El ejecutable anterior instala en el equipo, si no lo están ya, algunos paquetes necesarios y realiza un procedimiento denominado *bind mount*. Este método replica (monta) el sistema de archivos en una ruta determinada dentro del propio sistema operativo. De este modo, se crea la jaula en la ruta `/opt/phoenix-chroot/`, haciendo que cualquier cambio que se efectúe sobre los ficheros del disco se vea reflejado dentro de este directorio. Si todo

transcurre correctamente, la consola devuelve un mensaje confirmando que los directorios se han montado correctamente.

Dentro de la misma carpeta en la que se encuentra el *script* lanzado anteriormente existen otros dos:

- `unmount.sh`, que inicia el procedimiento de desmontaje de `phoenix-chroot` y debe lanzarse obligatoriamente antes de proceder a eliminar la instalación para evitar dañar el sistema de archivos.
- `after-boot.sh`, que es necesario ejecutar tras hacer uso de `unmount.sh` o después de haber reiniciado el equipo, puesto que los *bind mounts* no son persistentes.

5.3. Ejecución

Completada la instalación, los núcleos de red de *Open5GCore* están listos para ser lanzados. La ejecución de los mismos está centralizada en el *script* `ph_init.sh`. Este archivo es el encargado de crear los distintos espacios de nombres de red y las direcciones IP asociadas a cada uno de los elementos del núcleo de red definidos en los ficheros de configuración (consultar detalle en el apartado correspondiente). A continuación, inicia una sesión de la utilidad `tmux`, la cual permite abrir una ventana de terminal con múltiples pestañas.

El ejecutable `ph_init.sh` establece tantas pestañas como espacios de nombres de red existen, y dentro de cada pestaña se lanza un proceso de *phoenix*, si el espacio de nombre de red pertenece a una entidad o función de red del núcleo, u otro servicio, como es el caso de los espacios de nombres de red definidos para *MySQL* o *BIND*, un *software* servidor DNS.

Así pues, los comandos a utilizar para levantar el núcleo de red aparecen seguidamente. Debe lanzarse dos veces `ph_init.sh` ya que la primera ejecución crea los espacios de nombres de red y la sesión de `tmux`, y es la segunda la que permite ingresar a dicha sesión para ver e interactuar con la implementación del núcleo de red.

```
cd /opt/phoenix/tools/ph_init/
./ph_init.sh
./ph_init.sh
```

El atajo de teclado `Ctrl+B`, seguido de alguna de las siguientes teclas, permite el manejo de `tmux`:

- `N`, para avanzar a la siguiente pestaña o `P`, para volver a la anterior.
- `0-9`, para ir a la pestaña correspondiente.
- Flecha arriba o abajo para activar la instancia de terminal superior o inferior de una pestaña.
- `D`, para salir de la sesión de `tmux` dejándola en segundo plano.

Para detener la ejecución del núcleo de red, solo hace falta ejecutar el siguiente comando tras dejar en segundo plano la consola:

```
./ph_init.sh down
```

5.3.1. Primer arranque

Si bien el procedimiento que se acaba de exponer es el habitual, la primera vez que vaya a lanzarse un núcleo de red deben seguirse algunos pasos adicionales. En primer lugar, hay que generar un fichero `local.conf` en el directorio de configuración de *phoenix*, `cfg`. Este archivo contiene una variable `cfgdir` que el *script* `ph_init.sh` usa para lanzar una de las dos implementaciones disponibles (EPC o 5GC). Con los siguientes comandos se define el fichero indicado:

```
cd /opt/phoenix/cfg/
sudo cp 5g/local.conf.example local.conf
```

Se ha utilizado la plantilla para lanzar el núcleo de red 5G. Resulta conveniente editar el archivo (`sudo nano local.conf`) y añadir la siguiente línea al principio del mismo:

```
cfgdir="/opt/phoenix/cfg/legacy/"
```

Añadiendo esta línea se consigue un fichero de configuración útil para lanzar ambos núcleos de red. Si se desea arrancar el EPC, cuya configuración se encuentra en la carpeta `legacy`, basta con comentar con una almohadilla (`#`) la variable que apunta a la configuración del 5GC (carpeta `5g`). Alternativamente, para iniciar el núcleo de red 5G, únicamente hace falta dejar comentada la línea añadida al archivo en el paso previo.

Finalmente, con la primera ejecución de `ph_init.sh` algunos módulos de *phoenix* dejan de funcionar puesto que no pueden acceder al contenido de las bases de datos que requieren. Por ello, deben importarse las bases de datos al servidor *MySQL* desplegado dentro de la jaula *chroot*, utilizando para ello estos comandos en la pestaña `sql` de la sesión `tmux`:

```
cd $cfgdir/sql
./apply-databases.sh
```

Los comandos anteriores referidos a *MySQL* necesitan ser lanzados en la primera ejecución de ambas implementaciones (*Release* 3 y *Release* 5), ya que las bases de datos a las que accede cada uno de los núcleos de red son independientes. Si se han seguido todos los pasos, en este punto *Open5GCore* estará desplegado.

5.4. Configuración

El proceso de configuración de *Open5GCore* es, en cierto modo, opcional, puesto que tras completar la instalación es posible ejecutar inmediatamente el *sotware* y disponer de sendos núcleos de red funcionales. No obstante, existen ciertos aspectos cuya modificación resulta de interés para adaptar la implementación a las necesidades del usuario.

El funcionamiento de los núcleos de red depende del contenido de los ficheros ubicados en `/opt/phoenix/cfg/`. Como ya se ha avanzado en el apartado anterior, en esta ruta tienen relevancia dos directorios, `legacy` y `5g`, donde se encuentran los archivos de configuración y la asignación de direcciones IP de cada componente del EPC y el 5GC, respectivamente, además de las bases de datos que usan.

5.4.1. Arquitectura por defecto del EPC

Inicialmente, la *Release* 3 de *Open5GCore* instalada haciendo uso de la operación *chroot* y el *script* `ph_init.sh` se despliega de acuerdo con el diagrama de la Figura 19. En el mismo no se han incluido otros componentes que, pese a ejecutarse por defecto tras la instalación, no forman parte de un EPC en el sentido estricto (por ejemplo, el móudulo LWM2M que implementa comunicaciones máquina a máquina o el ANGW).

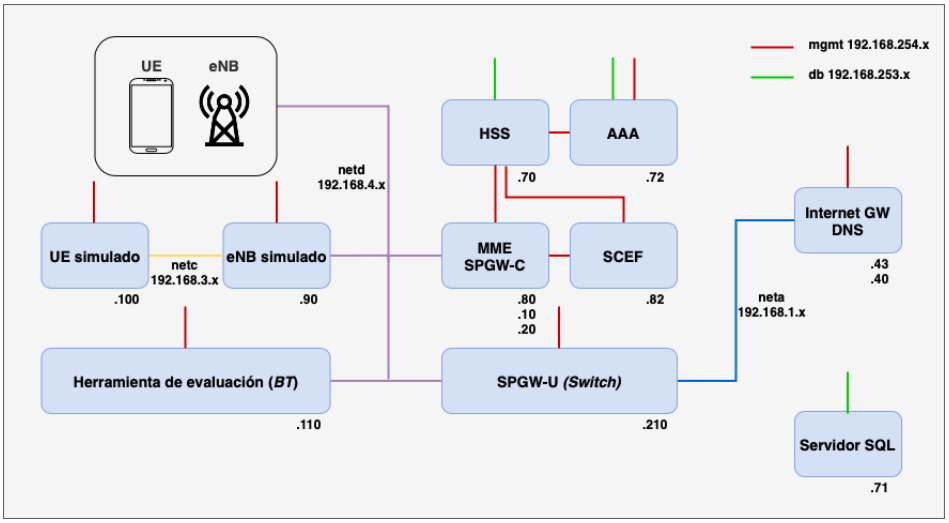

Figura 19: Arquitectura incial de la implementación del EPC de *Open5GCore*

Del diagrama cabe destacar los siguientes puntos:

- Las entidades ligadas al plano de control, es decir, el MME, el SGW-C y el PGW-C, se ejecutan en una única instancia de *phoenix*, igual que en el caso del SGW-U y el PGW-U, etiquetados como *Switch*, encargados del plano de usuario.
- El HSS y el servidor AAA constituyen dos entidades diferenciadas.
- El núcleo de red implementa el SCEF (*Service Capability Exposure Function*), cuya función es dar soporte a aplicaciones IoT.

- *Open5GCore Release* 3 incluye sendos módulos que emulan una estación base eNB y un UE, así como una herramienta de evaluación (*Benchmarking tool*) que simula tres eNB y hasta 1000 usuarios, permitiendo realizar diferentes pruebas y medidas.
- Se incluye un IGW (*Internet gateway* o puerta de enlace a Internet) para redireccionar el tráfico que proviene del plano de usuario y un servidor SQL.

La Figura 19 también recoge las distintas redes necesarias para interconectar las distintas entidades del núcleo de red, especificando sus nombres y direcciones IP, así como las de los componentes conectados a ellas:

- `neta`: esta red conecta el SPGW-U con los módulos dedicados a aplicaciones (no se abordan en este proyecto) y es la que proporciona acceso a Internet.
- `netc`: exclusiva para conectar el UE al eNB cuando se utilizan estos componentes emulados (en la realidad se correspondería con señales radio).
- `netd`: es la red sobre la que se despliega el protocolo GTP para transportar señalización y datos entre el eNB, el MME y el SPGW-U, por un lado, y para comunicar las funciones del plano de control y el plano de usuario del SGW y el PGW.
- `db`: las entidades que requieren acceso a bases de datos se conectan al servidor *MySQL* a través de esta red.
- `mgmt`: la comunicación entre el MME y el HSS para tareas de autenticación se realiza por esta red utilizando el protocolo *Diameter*; también sirve para establecer una conexión directa con cada una de las entidades.

La implementación de estas redes en el equipo que aloja la instalación y la conexión de cada uno de los espacios de nombres de red en los que se ejecutan los componentes a dichas redes se consigue por medio de *bridges*, etiquetados con el prefijo `br-` (así, `br-neta`, `br-mgmt`, etc.).

Para completar el detalle de la configuración que el EPC presenta por defecto, la Tabla 10 recoge los parámetros más relevantes definidos en las entidades de control del núcleo de red, los cuales deben coincidir en la RAN y/o la tarjeta SIM del UE para lograr un despliegue efectivo del prototipo de red móvil.

Parámetro	Valor
MCC	001
MNC	01
TAC	1
APN	default

Tabla 10: Códigos definidos en el EPC para la conexión de usuarios y estaciones base

5.4.2. Arquitectura por defecto del 5GC

Siguiendo el mismo desarrollo realizado para la *Release* 3, la *Release* 5 presenta en su despliegue inicial la arquitectura descrita en el diagrama de la Figura 20.

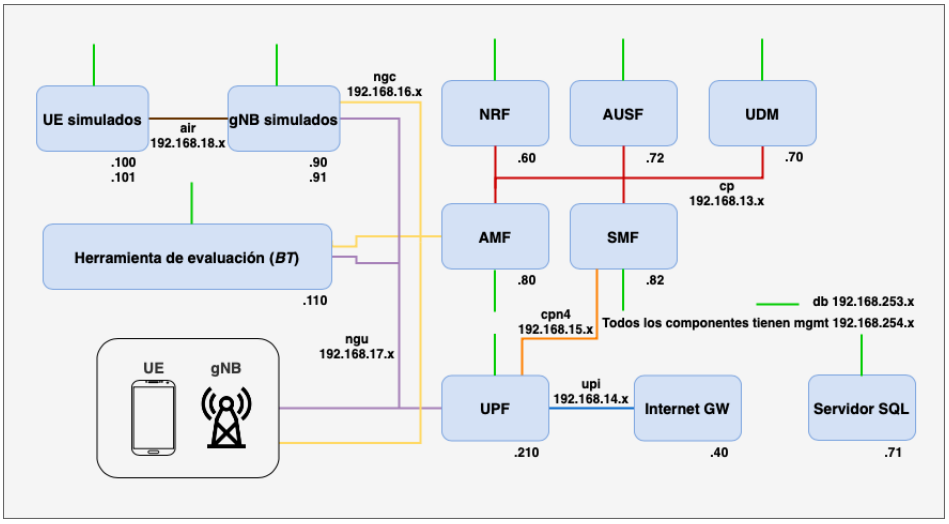

Figura 20: Arquitectura inicial de la implementación del núcleo de red 5G

Respecto a la Figura anterior son reseñables los siguientes aspectos:

- La implementación refleja uno de los principios en los que se basan los núcleos de red de 5G, con las funciones de red como servicios disponibles entre sí en el plano de control en vez de estar interconectadas mediante interfaces punto a punto.
- *Open5GCore Release* 5 también incluye una pareja de estaciones base gNB y usuarios emulados, y mantiene la herramienta de evaluación presente en la *Release* 3 adaptada a la 5G.
- Forman parte de la implementación el IGW y el servidor SQL con las mismas atribuciones que en el EPC.
- Pese a lanzarse por defecto un único UPF, se incluye otro archivo de configuración para poder desplegar de forma inmediata dos si se está interesado en ello.

Desde el punto de vista de las redes que forman parte de la implementación, el método seguido para su creación es el mismo que en la otra versión, y consiste en utilizar *bridges* virtuales para interconectar los diferentes espacios de nombres de red. En este contexto se han definido las siguientes redes, pudiendo consultarse las direcciones IP en la Figura 20:

- `air`: exclusiva para conectar el UE al gNB cuando se utilizan estos componentes emulados (en la realidad se correspondería con señales radio).
- `cp`: la señalización en el plano de control entre las distintas funciones de red ocurre a través de esta interfaz.
- `cpn4`: es la red que conecta el SMF con el UPF.
- `db`: las funciones de red que requieren acceso a bases de datos se conectan al servidor *MySQL* a través de esta red.

- `mgmt`: sirve para establecer una conexión directa con cada una de las funciones de red, además de ser la interfaz por la que el IGW envía el tráfico de usuario hacia Internet.
- `ngc`: conecta el gNB con el AMF.
- `ngu`: conecta el gNB con el UPF.
- `upi`: esta red canaliza el tráfico de usuario desde el UPF hacia el IGW.

Finalmente, la Tabla 11 recoge los parámetros de la Tabla 10 definidos por defecto en el 5GC.

Parámetro	Valor
MCC	001
MNC	01
TAC	117
APN	default

Tabla 11: Códigos definidos en el 5GC para la conexión de usuarios y estaciones base

5.4.3. Lanzamiento de componentes y ajustes de red

El fichero ip-map

Tal y como se ha visto, los núcleos de red se inician con una serie de módulos activos por defecto. Sin embargo, es posible desactivar aquellos cuyo uso no interesa e incluso añadir nuevos módulos configurados manualmente. El archivo que gestiona esto se denomina `ip-map` y cada *Release* tiene en su directorio de configuración uno propio. La estructura de este fichero puede observarse en la Figura 21.

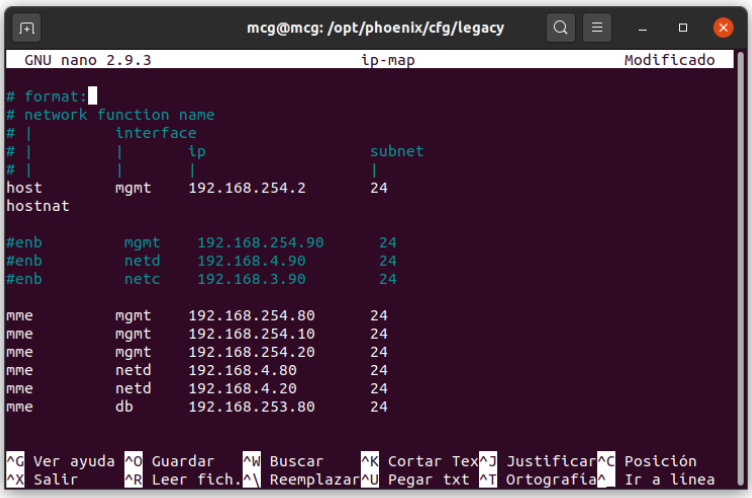

Figura 21: Fichero de configuración *ip-map* del EPC

En `ip-map` se incluyen todos los componentes implementados en el correspondiente núcleo de red, las interfaces a las que están conectados y la dirección IP asignada dentro de la subred. En cada línea se anota un espacio de nombre de red, que coincide con la entidad o función de red que se va a ejecutar dentro del mismo, la red a la cual se le proporciona conexión y la dirección IP que usa en dicha red, acompañada de la máscara de subred debidamente ajustada para que el espacio de nombre de red pueda conectarse al resto de nodos de la red. Se repiten tantas líneas con el mismo espacio de nombre de red como interfaces de conexión se le desea proporcionar.

Elección de los componentes que se ejecutan

Los distintos componentes disponibles en *Open5GCore* se lanzan como un proceso de *phoenix* aislado en un espacio de nombre de red si existe en el directorio definido para cada núcleo de red (`legacy` y `5g`) un archivo de configuración con el mismo nombre en formato XML o JSON. No obstante, es posible omitir un espacio de nombre de red determinado si se comentan con una almohadilla (`#`) las entradas referidas al mismo en el archivo `ip-map`.

Por otro lado, si se desea establecer un nuevo componente (por ejemplo, un MME o un UPF adicionales), los pasos a seguir son crear un nuevo fichero de configuración de *phoenix* en la carpeta correspondiente y añadir una o varias entradas asociadas al mismo en `ip-map`, dependiendo del número de interfaces de red que se le doten.

Los cambios surten efecto tras reiniciar la ejecución de los núcleos de red.

Modificación de las interfaces de red

La dirección de las redes existentes en cada implementación es otro aspecto sujeto a cambios. Para ello, el fichero `ip-map` también es el punto de partida. Así pues, deben editarse en dicho fichero aquellas direcciones que se busque renovar, teniendo en cuenta que al cambiar la dirección de red de uno de los componentes, necesariamente ha de realizarse la misma modificación en el resto de componentes de la red para que no pierdan la conexión. Asimismo, debe revisarse que la nueva dirección de red no esté siendo usada por otra interfaz en el equipo.

A continuación, se desarrolla el caso del cambio de dirección IP de la red `netd` a `200.x.y.z`. El primer paso es eliminar el *bridge* creado en el equipo para interconectar los equipos de dicha red. Para ello, se ejecutan los siguientes comandos:

```
sudo ip link set dev br-netd down
sudo brctl delbr br-netd
```

Después hay que modificar las direcciones asociadas a la interfaz `netd` de la que disponen todos los espacios de nombres de red conectados a la red del mismo nombre. Obsérvese que la dirección IP propuesta para la red viene definida por una máscara de red de 8 bits, con lo que se dispone de más direcciones en la subred. Por tanto, también debe modificarse este parámetro en las correspondientes entradas. Por ejemplo:

```
dpsw    netd    200.2.2.20    8
```

Tras ello, si la alteración de las direcciones IP afecta a una red implementada en la *Release* 3, hay que actualizar el archivo de zona de DNS para que refleje los cambios. Este

paso no es necesario en la *Release* 5 puesto que dicha versión no recurre a un servidor DNS para resolver los FQDN de cada componente del núcleo de red.

La ruta del archivo de zona de DNS es `cfg/open5gcore.zone`. En este fichero, aparecen los diferentes dominios con el formato `<espacio de nombre de red>.<red>`. Por tanto, la entrada modificada anteriormente tendría el siguiente aspecto en este archivo:

```
dpsw.netd       IN A 200.2.2.20
```

Finalmente, también deben corregirse las referencias a las direcciones IP modificadas presentes en los archivos de configuración de los módulos de *phoenix* ubicados en `cfg/legacy/` o `cfg/5g/` (según el núcleo de red que se esté configurado).

Los cambios anteriores pueden automatizarse por medio del siguiente comando ejecutado en la carpeta `cfg`, en el que se indica la dirección IP antigua `192.168.4.x` y la nueva `200.x.y.z` (el comando se lanza en una única línea):

```
cd /opt/phoenix/cfg/
fgrep -rIsl --null '192.168.4.x' .| xargs -r -0 sed -i
 's|192.168.4.x|200.x.y.z|g'
```

Completados todos los pasos, basta con ejecutar `ph_init.sh` para reconstruir el *bridge* eliminado al principio y lanzar el núcleo de red con las direcciones IP de los espacios de nombres de red actualizados.

Configuración del DNS para permitir tráfico hacia Internet

Open5GCore hace uso de los servicios DNS del equipo en el que se ha llevado a cabo la instalación para resolver los nombres de dominio requeridos por el usuario conectado a la red. Por ello, si se experimentan problemas de conexión, como imposibilidad de cargar páginas web, probablemente tienen relación con el DNS.

En Linux, el archivo que gestiona la resolución DNS es `resolv.conf` y se ubica en `/etc/`. Este fichero debe editarse para incluir un servidor DNS que responda a las peticiones del usuario. Puede establecerse la dirección IP del DNS de la red a la que está conectada la máquina o, por ejemplo, la de Google (`8.8.8.8`). Se añade la siguiente línea:

```
nameserver x.y.z.w
```

5.4.4. Actualización del PLMN

El PLMN (*Public Land Mobile Network*) es el código que identifica de forma unívoca a un operador de red dentro de un país. Está compuesto, por tanto, por el MCC y MNC. Por defecto, los núcleos de red de *Open5GCore* tienen asignados los códigos 001 y 01, tal y como indican las Tablas 10 y 11.

No obstante, puede llegar a resultar necesario cambiarlos ya que el MCC y el MNC del núcleo de red y los cinco primeros dígitos del IMSI registrado en las tarjetas SIM,

correspondientes a estos parámetros, deben coincidir. En caso contrario dichos usuarios no podrán tener acceso a la red desplegada. Del mismo modo, el MCC y el MCC con los que trabajan las estaciones base también debe ser igual al configurado en el núcleo de red.

El PLMN aparece en diferentes ubicaciones: los ficheros de configuración de cada uno de los componentes y, en la *Release* 3, los ficheros de configuración del DNS y las tablas de las bases de datos. El MCC y el MNC aparece de manera individual en los primeros, mientras que en todos ellos figura también dentro del FQDN de cada uno de los componentes.

Para modificar de forma automática todas las referencias al PLMN pueden ejecutarse los siguientes comandos, mediante los cuales se actualiza el MCC y el MNC a 208 y 93, respectivamente. Se hace coincidir con los incluidos en el IMSI de la tarjeta USIM programada en el Capítulo 3 (ver Tabla 3).

```
cd /opt/phoenix/cfg/
fgrep -rIsl --null 'mnc001.mcc001.3gppnetwork.org' .| xargs -r -0 sed
 -i 's|mnc001.mcc001.3gppnetwork.org|mnc093.mcc208.3gppnetwork.org|g'

fgrep -rIsl --null 'mnc="1"' .| xargs -r -0 sed
 -i 's|mnc="1"|mnc="093"|g'

fgrep -rIsl --null 'mcc="1"' .| xargs -r -0 sed
 -i 's|mcc="1"|mcc="208"|g'

cd /opt/phoenix/tools/
fgrep -rIsl --null 'mnc001.mcc001.3gppnetwork.org' .| xargs -r -0 sed
 -i 's|mnc001.mcc001.3gppnetwork.org|mnc093.mcc208.3gppnetwork.org|g'
```

Los comandos `fgrep` anteriores se ejecutan en una única línea. Nótese también que el MNC se ha rellenado con un 0 inicial. En principio, el MNC es un código de dos o tres dígitos, pero *Open5GCore* toma los códigos precedidos de ceros como el mismo MNC.

Se puede observar que se ha actualizado asimismo el dominio en los archivos localizados en la ruta `tools/`. En este directorio se ubican diversos *scripts* que se utilizan para modificar el contenido de la base de datos del HSS y los ficheros de configuración del servicio de DNS *BIND*. Estos últimos se copian en el directorio `/etc/bind/` de la jaula *chroot* para que la resolución de nombres siga funcionando correctamente:

```
cd ph_init/chroot/etc/
sudo cp bind/* /opt/phoenix-chroot/etc/bind/
```

5.4.5. Registro de usuarios en la base de datos del HSS

El registro de usuarios es uno de los aspectos más importantes en el proceso de configuración del núcleo de red. Este procedimiento es el que posibilita que los usuarios puedan autenticarse en la red móvil y conectarse a ella. El elemento que almacena los datos de todos los usuarios es la base de datos del HSS, que por defecto viene cargada con un millar

de usuarios utilizados por el UE simulado y la herramienta de evaluación.

La agregación de usuarios es un proceso automatizado gracias a los *scripts* disponibles en el directorio `tools/hsstools/`. Si no se indica lo contrario, los pasos descritos a continuación deben realizarse con el EPC en ejecución, ya que se lanzarán los comandos en el espacio de nombre de red del servidor *MySQL*. De hacerlo fuera de este entorno, podrían dañarse las bases de datos del servidor *MySQL* del equipo, si este existe.

En primer lugar, se recomienda restablecer la base de datos del HSS. Borrar los datos registrados implica perder la funcionalidad del equipo de usuario emulado y de la herramienta de evaluación. A cambio, después de completar el proceso podrán registrarse tantos usuarios como se desee con la información de sus respectivas tarjetas SIM.

Para limpiar la base de datos actual se ejecuta el siguiente comando, aportando el MCC (`-c`) y el MNC (`-n`) establecido previamente en el núcleo de red y codificado en el IMSI de los usuarios:

```
cd /opt/phoenix/tools/hsstools/
./prepare_db.sh -n 93 -c 208
```

```
(sql)root@mcg:/opt/phoenix/tools/hsstools# ./prepare_db.sh -n 93 -c 208
Parameter ims is yes
Applying hss_db_clean.sql
Adding Visited Network mnc093.mcc208.3gppnetwork.org
Adding Default QoS Profiles
Adding Default APN Configuration Profile
Adding Default APN Configurations
Adding Default APN Configurations to Profile mappings
Adding Default Data for Sp Clients
Adding Visited Network ims.mnc093.mcc208.3gppnetwork.org
Adding QoS Profile for VoLTE
Adding APN Configuration for IMS
Adding Application Server
Adding Preferred S-CSCF Set
COUNTRY is 49
Added 1 users to the HSS database
Database prepared!
(sql)root@mcg:/opt/phoenix/tools/hsstools#
```

Figura 22: Base de datos del HSS de *Open5GCore* restablecida

Si todo transcurre sin errores aparecerán mensajes como los de la Figura 22, en el que se puede observar el dominio con los códigos MCC y MNC actualizados. Entonces se procede a añadir el usuario con los datos de la Tabla 3:

```
./provision.sh -u user1 -I 208930000000003 -M 000002
-n mnc093.mcc208.3gppnetwork.org -k 8baf473f2f8fd09487cccbd7097c6862
-o 11111111111111111111111111111111 -s 128 -v yes
```

En el comando anterior también debe utilizarse el MSISDN y el SPN proporcionados por la herramienta de programación de las tarjetas USIM (Figura 13). Con la opción `-u` se proporciona un nombre de usuario único para su identificación en la base de datos, y con la opción `-o` se informa del código OP que utiliza el HSS y con el que debe programarse la tarjeta.

Se comprueba que el usuario se ha añadido correctamente si tras la ejecución del *script* se muestran los mensajes de la Figura 23.

Por último, es posible registrar de una sola vez una cantidad determinada de usuarios si estos comparten la clave de autenticación (Ki) y el parámetro OP. Para ello, primero deben modificarse los valores de los correspondientes campos en el *script* `provision.sh`:

```
K="(UNHEX(NULL))"
OP="(UNHEX('00000000000000000000000000000000'))"
```

A su vez, en el *script* `mass_provision.sh` se define el IMSI y el MSISDN iniciales:

```
imsi_suffix="<IMSI>"
msisdn_suffix="<MSISDN>"
```

Tras ello, se ejecuta este último archivo con las siguientes opciones desde el espacio de nombre de red `sql`. Aparecerán tantos mensajes como los de la Figura 23 como usuarios se hayan añadido.

```
cd /opt/phoenix/tools/hsstools/
./mass_provision.sh -n 93 -c 208 -u <número_usuarios>
```

```
Parameter ims is yes
@id_imsi := last_insert_id()
6
@imsu := last_insert_id()
6
@id_msisdn := last_insert_id()
6
@impi := last_insert_id()
6
@impu := last_insert_id()
6
User user1 is created !
(sql)root@mcg:/opt/phoenix/tools/hsstools#
```

Figura 23: Usuario registrado en la base de datos del HSS de *Open5GCore*

5.4.6. Conexión de estaciones base eNB/gNB físicas

La conexión de una RAN alternativa a los componentes emulados de *Open5GCore* requiere dar visibilidad a la red que conecta los núcleos de red a los eNB y/o gNB, es decir, asignar una dirección IP al *bridge* que sirva de puerta de enlace. Así, será posible establecer comunicación con los espacios de nombres de red implicados (MME, SPGW-C y SPGW-U en el EPC, AMF y UPF en el 5GC) desde el propio equipo o desde otras máquinas.

En el 5GC este paso no es necesario, pues las redes `ngc` y `ngu` ya tienen una puerta de enlace. En el caso del EPC, debe añadirse a su fichero `ip-map` la línea referida a la red

`netd` al principio del mismo, como se observa en la Figura 24. Las entradas de este tipo permiten conectarse a las redes implicadas desde el exterior.

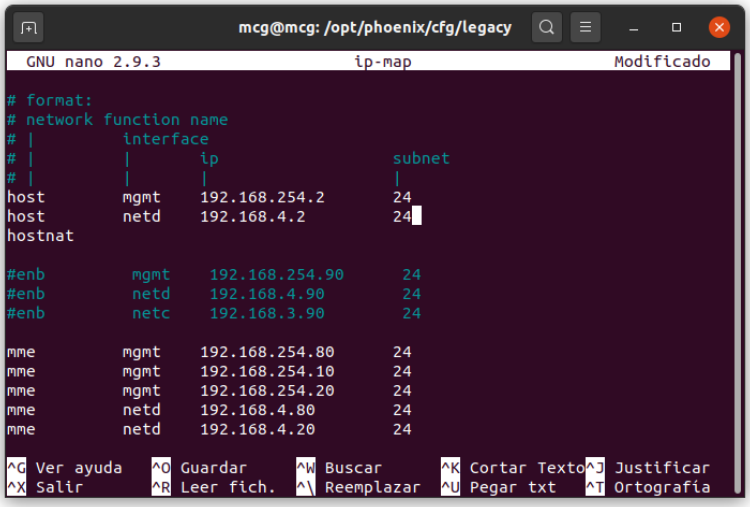

Figura 24: Configuración de red para aceptar conexiones de estaciones base físicas

Si la RAN se ejecuta en el mismo equipo que *Open5GCore* (Figura 8) bastará con conectar el eNB o gNB al MME o AMF según las direcciones IP de la Tabla 12, puesto que las reglas de encaminamiento del sistema son suficientes para transmitir los paquetes de datos. Si la RAN está desplegada en otra estación de trabajo (Figura 9), adicionalmente es necesario incluir la interfaz o interfaces de red *Ethernet* a través de las cuales se establece conexión con dicho equipo en los *bridges* correspondientes. Suponiendo que se utiliza una interfaz denominada `eth1`:

```
sudo brctl addif br-<nombre_red> eth1
```

MME	AMF
192.168.4.80	192.168.16.80

Tabla 12: Direcciones IP por defecto del MME y el AMF de *Open5GCore*

Por último, la dirección IP de la interfaz o interfaces usadas por el segundo equipo debe formar parte de la misma subred que los componentes mencionados anteriormente.

Capítulo 6

Diseño del escenario de pruebas

Tras el despliegue de los núcleos de red implementados por las plataformas *OpenAirInterface* y *Open5GCore*, en este capítulo se explica el procedimiento propuesto para probar dichas soluciones. Este procedimiento se ha diseñado en torno a dos fases: la validación de los núcleos de red y la evaluación de sus prestaciones.

Mediante la validación se pretende comprobar el funcionamiento de los núcleos de red, desplegando el prototipo de red móvil completo con el objetivo de verificar que los usuarios pueden inician sesión y acceder a la red IP externa conectada al sistema móvil. Por otro lado, se lleva a cabo una evaluación de sus prestaciones a partir un conjunto de casos de uso definidos y ejecutados con la herramienta *Spirent Landslide*. De este modo, se obtiene una serie de resultados objetivos útiles para comparar el rendimiento de ambas soluciones de forma certera.

6.1. Validación de las soluciones propuestas

El proceso de validación de los núcleos de red comprende su integración en dos escenarios. El primero es el escenario simulado, en el que se establece la red al completo utilizando una red de acceso radio (RAN) y un equipo de usuario (UE) emulados. El segundo escenario es el real, donde los núcleos de red se integran con una RAN física implementada en el laboratorio mediante un equipo SDR y el *software* que gobierna dicho equipo para funcionar como estación base.

6.1.1. Integración de los núcleos de red: escenario simulado

Las dos plataformas utilizadas en este proyecto incluyen componentes que simulan estaciones base físicas a la vez que usuarios para probar la conexión al núcleo de red. En el caso de *Open5GCore*, estos elementos forman parte de la instalación junto al resto de componentes del EPC y el 5GC. Sin embargo, OAI distribuye el simulador como parte del módulo dedicado a la implementación *software* de la RAN.

Dado que el despliegue y la utilización de dicho paquete no forman parte de los objetivos de este proyecto, únicamente se plantea la integración con la parte radio simulada de los

núcleos de red de *Open5GCore*. El lector puede revisar en el Capítulo 5 dedicado a esta plataforma los componentes que pueden utilizarse en el diseño del escenario simulado. Son, principalmente, los módulos de *phoenix* que implementan estaciones base eNB y gNB y usuarios con conectividad LTE y 5G, así como las respectivas herramientas de evaluación (*Benchmarking Tools*) de cada uno de los núcleos de red.

Para la realización de las pruebas que se proponen, se ha recurrido a los componentes autónomos. Además, estas pruebas se han llevado a cabo con la configuración por defecto de *Open5GCore* para aprovechar que los parámetros de los UE simulados están registrados en la base de datos del HSS y el UDM. Con ello se evita tener que modificar estas bases de datos y los ficheros de configuración de los componentes utilizados.

Una limitación de esta prueba que debe mencionarse es la imposibilidad de simular un escenario 5G NSA, de manera que solo se procede con la conexión al EPC de un usuario con conectividad LTE a través de un eNB. Por su parte, al 5GC se conecta un usuario con conectividad 5G por medio de un gNB que gestiona tanto el plano de control como el de usuario, es decir, en una configuración SA.

6.1.2. Integración de los núcleos de red: escenario real

La segunda parte de la validación de los núcleos de red consiste en desplegar una RAN a través de la cual los UE comerciales disponibles para el desarrollo de este proyecto, haciendo uso de la tarjeta USIM programada, puedan conectarse a la red. La RAN que se utiliza en este escenario se compone de un dispositivo SDR, el USRP B210 descrito en el Capítulo 3, controlado por el *software* de OAI que implementa el acceso radio LTE (4G) y NR (5G). De esta manera puede disponerse de eNB y gNB físicos.

La principal restricción de este escenario es en realidad una limitación de la implementación de la RAN desarrollada por OAI. En el momento de la realización del proyecto, la RAN solo admite una configuración de red 5G NSA [28], de modo que no puede plantearse su integración con la *Release* 5 de *Open5GCore* (5GC).

Asimismo, el *software* para el despliegue de gNB se encuentra en estos momentos en fase de desarrollo y no se garantiza su correcto funcionamiento. Por ello, no es posible validar con garantías los núcleos de red LTE en un escenario 5G NSA con los recursos disponibles para esta prueba. En definitiva, la conexión de los dispositivos móviles a la red en este escenario solo se garantiza si se despliega un eNB; por tanto, la red resultante es LTE.

Conexión del eNB a los núcleos de red

La integración del eNB de OAI con las soluciones de EPC desplegadas es un proceso en el que deben coincidir ciertos parámetros para establecer la conexión de manera satisfactoria. En primer lugar, tanto si todas las partes de la red están instaladas en un mismo equipo como si lo están en dos, debe configurarse una dirección IP en el eNB que habilite la conexión al MME del EPC.

Preferiblemente, el eNB y el MME deben estar en la misma subred, evitando de este modo recurrir a las reglas de encaminamiento del sistema. La prueba se realiza sobre un único equipo utilizando la configuración del núcleo de red de OAI propuesta para dicho

escenario. Respecto a la *Release* 3 de *Open5GCore*, se emplea la configuración de red por defecto y las indicaciones sobre conexión de eNB/gNB físicos recogidas en el apartado de configuración del despliegue de dicho núcleo de red. Por tanto, la dirección IP del MME en ambos casos es:

- **OAI:** 172.168.3.17/8
- ***Open5GCore*:** 192.168.4.80/24

Por otra parte, también deben coincidir los parámetros que identifican el núcleo de red. Estos son el MCC, el MNC y el TAC, presentados a lo largo del documento. Los códigos que se utilizan, modificados respecto a la la configuración por defecto del MME del EPC de *Open5GCore* para que el UE que haga uso de la tarjeta USIM programada pueda autenticarse en la red, son los siguientes:

- **MCC:** 208
- **MNC:** 93
- **TAC:** 600 (OAI), 1 (*Open5GCore*)

Ajuste del APN en el dispositivo móvil

Finalmente, tras introducir la tarjeta USIM programada en el UE que se conecta a la red, debe añadirse el APN asociado al usuario en la base de datos del HSS. El menú desde el que se modifica dicho parámetro suele ubicarse, por lo general, bajo los ajustes de redes móviles en dispositivos Android e iOS. Los APN configurados en cada núcleo de red son:

- **OAI:** oai.ipv4
- ***Open5GCore*:** default

Configurados todos los parámetros anteriores, se está en condiciones de desplegar el prototipo completo de red LTE y probar la conexión de los dispositivos móviles comerciales disponibles en el laboratorio para este proyecto.

6.2. Evaluación del rendimiento de los núcleos de red

Las pruebas de conexión de un UE a los núcleos de red sirven para certificar que la configuración propuesta para sus despliegues es adecuada y que, por tanto, se cumple el objetivo de disponer de redes funcionales. Sin embargo, estos escenarios son muy sencillos y no reflejan la complejidad de una red móvil en la realidad, en la que múltiples estaciones base se conectan al núcleo de red y ofrecen servicio a cientos de usuarios.

Por tanto, es importante llevar a cabo pruebas más exigentes para descubrir las prestaciones reales de los núcleos de red y, de esta manera, poder extraer conclusiones objetivas y bien fundamentadas acerca de su rendimiento. No obstante, simular un escenario de estas características en un entorno de laboratorio es una tarea difícil porque los recursos físicos

son limitados. Por ejemplo, no es verosímil plantear ensayos con cientos de *smartphones* como UE.

Para dar respuesta a esta problemática, surgen las herramientas de evaluación del rendimiento de los sistemas de comunicaciones móviles, que ofrecen la posibilidad de poner a prueba todas las partes de una red móvil recreando situaciones del mundo real. Estas soluciones gozan de amplia difusión en el sector por su versatilidad y son utilizadas tanto en el ámbito académico y de investigación como entre los proveedores para el desarrollo de nuevos componentes de las redes.

La evaluación de las prestaciones de los núcleos de red de OAI y *Open5GCore* en el laboratorio se efectúa mediante la herramienta *Spirent Landslide*, de la que el MCG ha adquirido una licencia de uso.

6.2.1. Acerca de *Spirent Landslide*

Spirent Landslide [29] es un completo sistema desarrollado por Spirent Communications para la prueba de extremo a extremo de multitud de redes de comunicaciones, entre ellas los sistemas de comunicaciones móviles de tercera a quinta generación, WiFi o IMS (*IP Multimedia Subsystem*).

Landslide ofrece a los usuarios una metodología que aúna componentes emulados y una variedad de casos definidos a partir de la cual se configura un entorno completamente controlado de pruebas. Se otorga así a investigadores y proveedores de equipos *hardware* y servicios la posibilidad de evaluar desde la red completa hasta nodos específicos de la misma [30].

Para realizar dichas pruebas, *Landslide* emula desde miles hasta millones de suscriptores móviles (en función de la licencia adquirida) que acceden de forma simultánea o gradual a la red. Estas redes pueden estar desplegadas físicamente mediante equipos *hardware* de fabricantes, virtualizadas en servidores de propósito general o basadas en la nube, de modo que los núcleos de red de última generación son compatibles con la herramienta.

Spirent Communications despliega *Landslide* sobre diferentes plataformas, dependiendo del tipo de usuario que adquiere la licencia. Para grandes corporaciones se ofrecen equipos físicos que garantizan la emulación de volúmenes elevados de tráfico, mientras que para el entorno de investigación y para la prueba de redes en la nube se oferta una versión *software*, lista para instalar en máquinas virtuales en equipos de propósito general.

Arquitectura del sistema

En cualquiera de las variantes disponibles, un despliegue de *Landslide* consta de tres componentes [31]:

- **TAS (*Test Administration Server*) o *Test Manager*:** es el módulo principal de *Landslide*, encontrándose entre sus funciones la gestión de las cuentas de usuario, la configuración general del sistema, el control de las sesiones de pruebas y la entrega de los resultados de las mismas al usuario.
- **TS (*Test Server*):** es el nodo del sistema al que se conectan los SUT (*System Under Test*) y en el que se ejecutan las pruebas, pudiendo desplegarse hasta 32 bajo

la supervisión de un único TAS.

- **Cliente *Landslide*:** es la interfaz gráfica de usuario, a través de la cual el usuario interacciona con el TAS para gestionar el sistema, configurar las sesiones de pruebas y visualizar los resultados de estas.

La Figura 25 es un diagrama de la arquitectura de *Landslide*, en la que pueden observarse los componentes anteriores.

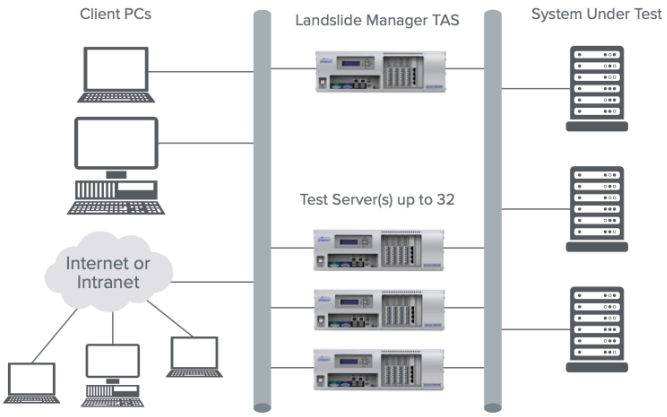

Figura 25: Arquitectura de *Spirent Landslide* [31]

6.2.2. Integración de los núcleos de red con *Spirent Landslide*

En el laboratorio del MCG, los componentes de *Landslide*, un TAS y un TS, están desplegados mediante máquinas virtuales sobre un hipervisor *VMWare* en un servidor igual al que aloja los núcleos de red implementados en este proyecto. Dicho hipervisor no precisa de un sistema operativo, por lo que los componentes disponen para sí mismos de todos los recursos *hardware* del equipo.

Con esta configuración, para probar los núcleos de red es necesario conectar físicamente los servidores. Dicha conexión se realiza a través de un *router* que asigna una dirección IP estática a las interfaces *Ethernet* utilizadas. Además, se configuran dos redes de área local virtuales (VLAN). Una se utiliza para la gestión del TS por parte del TAS y la segunda, cuya subred es `200.0.0.0/8`, viene a ser la red en la que se ubicarán los sistemas a probar.

El puerto del que dispone la máquina virtual del TS para conectar los sistemas a probar tiene asignada una dirección IP en el rango anterior. Se define una máscara de subred de 8 bits para no tener limitaciones en el número de direcciones IP disponibles, ya que todos los nodos que intervienen en la prueba (red o componente bajo prueba, componentes emulados...) deben estar conectados a la misma red.

De este modo, para conectar los núcleos de red al puerto de pruebas del TS, es necesario crear en la estación de trabajo una interfaz o *link* de dicha VLAN sobre la interfaz *Ethernet* utilizada para la conexión del equipo con el *router*.

Para la sesión de pruebas de un núcleo de red, *Landslide* emula la RAN. Por ello, de la misma manera que en el despliegue de la red móvil con un eNB físico, deben configurarse

correctamente las direcciones IP de las interfaces a través de las cuales la estación base se conecta al núcleo de red, teniendo en cuenta que, a diferencia del caso anterior, ahora se adapta la configuración de los núcleos de red.

A continuación se expone el procedimiento a seguir suponiendo que se utiliza la interfaz `eno2` del equipo para la conexión al *router* y a *Landslide*. En el caso del núcleo de red de OAI, se escoge la dirección IP `200.16.6.3/8`. Esta interfaz y dirección IP deben utilizarse para la interfaz S1-C del MME y la interfaz S1-U del SPGW-U.

Para crear una interfaz VLAN `eno2.2` se utiliza el siguiente comando:

```
sudo ip link add link eno2 name eno2.2 type vlan id 2
```

Para asignarle la dirección IP anterior, el comando es:

```
sudo ip addr add 200.16.6.3/8 dev eno2.2
```

Por último, para activar la interfaz VLAN creada se emplea el siguiente comando:

```
sudo ip link set dev eno2.2 up
```

Tras estos pasos, el EPC de OAI es visible para *Landslide* y puede empezar a probarse siempre que los usuarios emulados por el sistema estén registrados en la base de datos del HSS.

En las implementaciones de núcleos de red de *Open5GCore*, el procedimiento es similar. En primer lugar debe modificarse la dirección de red de los componentes que tienen conexión con la RAN, de acuerdo con las indicaciones del apartado de configuración del Capítulo dedicado al despliegue de la plataforma.

Seguidamente, se crea y se activa la interfaz VLAN utilizando los comandos anteriores, pero en este caso no se le asigna una dirección IP, sino que se añade al *bridge* correspondiente al que se conectan las estaciones base, tal y como se explica en la mencionada sección.

Capítulo 7

Análisis de resultados

En este capítulo se muestra la conexión de un usuario a la red móvil en los escenarios planteados en el Capítulo 6. A continuación, se presentan las pruebas definidas en *Spirent Landslide* para la evaluación de las prestaciones de inicio de sesión y carga de usuarios en la red de los EPC de OAI y *Open5GCore*. El objetivo de esta evaluación es comparar el rendimiento de ambas soluciones a partir de una serie de resultados objetivos.

7.1. Conexión de un UE simulado a *Open5GCore*

La arquitectura inicial de los núcleos de red que implementa *Open5GCore* ofrece una serie de componentes que simulan equipos de usuario (UE) y el acceso radio LTE y NR. Se pretende comprobar que, en las condiciones iniciales, estos UE pueden conectarse a la red y acceder a Internet.

Conexión del UE emulado al EPC

Para realizar esta prueba se lanza la *Release* 3 de *Open5GCore* con los componentes `ue` y `enb`. De manera inmediata puede observarse en la pestaña del MME el intercambio de señalización entre este y el eNB. También se puede utilizar el comando `mme.enb.print` con el mismo propósito.

A continuación, desde la pestaña definida para el UE se ejecuta el siguiente comando para conectar al usuario a la red. Si el UE se vincula con éxito a la red, en la consola del MME aparecen los mensajes de la Figura 26.

```
ue.connect_l3 1 LTE-N1
```

Regresando a la pestaña del UE, se envían mensajes ICMP mediante la utilidad `ping` al servidor en el que se aloja la página de inicio de Google para verificar la conexión a Internet. El resultado satisfactorio se muestra en la Figura 27, comprobando de paso el correcto funcionamiento del servicio DNS si se han seguido los pasos descritos en la sección de configuración de red de *Open5GCore*.

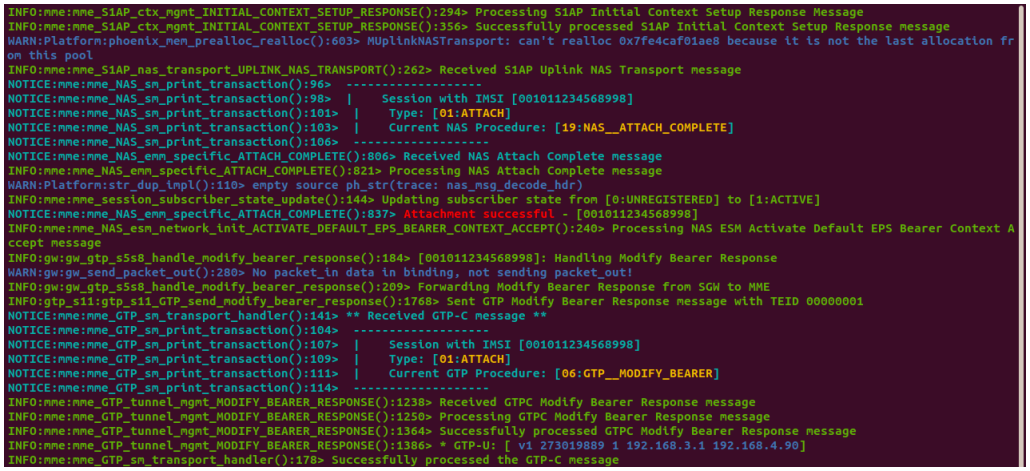

Figura 26: UE simulado conectado al EPC de *Open5GCore*

```
root@mcg:~# ip netns exec ue  bash
root@mcg:~# ping www.google.com
PING www.google.com (216.58.211.36) 56(84) bytes of data.
64 bytes from mad08s05-in-f4.1e100.net (216.58.211.36): icmp_seq=1 ttl=115 time=7.62 ms
64 bytes from mad08s05-in-f4.1e100.net (216.58.211.36): icmp_seq=2 ttl=115 time=7.91 ms
64 bytes from mad08s05-in-f4.1e100.net (216.58.211.36): icmp_seq=3 ttl=115 time=7.74 ms
64 bytes from mad08s05-in-f4.1e100.net (216.58.211.36): icmp_seq=4 ttl=115 time=7.76 ms
64 bytes from mad08s05-in-f4.1e100.net (216.58.211.36): icmp_seq=5 ttl=115 time=7.73 ms
64 bytes from mad08s05-in-f4.1e100.net (216.58.211.36): icmp_seq=6 ttl=115 time=7.96 ms
```

Figura 27: *Ping* a *www.google.com* desde el UE simulado

Al desconectar el UE de la red, la consola del MME muestra mensajes similares a los de la Figura 26 en sentido inverso. Se utiliza el siguiente comando para la desconexión:

```
ue.disconnect_l3 1 LTE-N1
```

Conexión del UE emulado al 5GC

Procediendo de forma análoga al caso anterior, se ejecuta la *Release* 5 de *Open5GCore* con los distintos componentes `ue` y `gnb`. En la consola del AMF pueden consultarse los gNB registrados lanzando el comando `amf.ng.print_nodes`. Por otro lado, en la pestaña de alguno de los UE cargados se utiliza el siguiente comando para conectarlo a la red:

```
ue_5g_nas_only.registration_and_pdu_connection
```

La Figura 28 muestra el proceso de autenticación del usuario en el núcleo de red que se desarrolla en el UDM. De la misma manera que en el caso anterior, se realiza la prueba de `ping`, obteniéndose resultados satisfactorios y similares a los conseguidos en la Figura 27. Finalmente, se desconecta el UE mediante el comando siguiente:

```
ue_5g_nas_only.deregistration_request
```

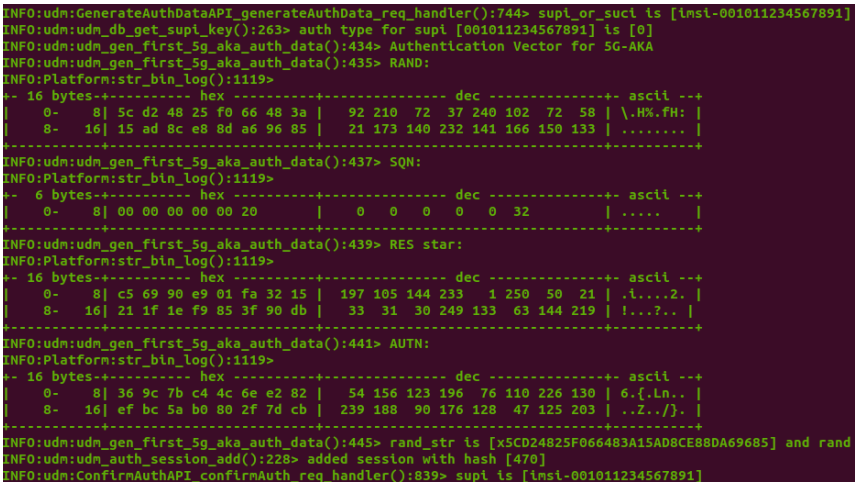

Figura 28: Proceso de autenticación del UE en el UDM

7.2. Conexión de un UE comercial a los núcleos de red LTE

Con esta prueba se quiere verificar que los EPC se integran correctamente con un eNB físico y ambos elementos están configurados correctamente para admitir la conexión de un usuario a través de los dispositivos móviles disponibles.

Una vez que se lanza la implementación del eNB y este establece conexión con el núcleo de red, puede iniciarse la conexión del UE. Con la tarjeta SIM insertada en el mismo, se selecciona el APN correspondiente a cada EPC y tras ello el usuario se conecta al prototipo de red móvil, pudiendo consultar en el dispositivo la dirección IP que le asigna el SPGW-C. En las consolas de los núcleos de red puede revisarse el intercambio de mensajes de señalización que tiene lugar entre el usuario, la estación base y el MME.

En la Figura 29 se muestra este intercambio de mensajes, pudiendo identificar el IMSI del usuario, que es el registrado en la tarjeta USIM, así como el túnel de comunicación para el plano de usuario (GTP-U) que se establece entre el UE, cuya dirección IP es `192.168.3.1`, y el eNB, configurado con la dirección `192.168.4.8`.

Mensajes similares aparecen en la ventana de terminal en la que se ejecuta el MME del EPC de OAI, donde destaca el cuadro de estadísticas de la Figura 30 que recoge los eNB y UE con sesión establecida.

En esta solución se ha podido comprobar que, a pesar de utilizar siempre la misma tarjeta USIM cuyos parámetros están registrados en la base de datos del HSS, la red rechaza la conexión del *smartphone* Galaxy A90 5G de Samsung. Tras investigar el por qué de dicho rechazo, se ha encontrado que el motivo por el que se deniega la conexión se debe a algunas licencias del fabricante que OAI no incluye.

Asimismo, un fallo detectado en el *software*, que puede observarse, es la no inclusión de las *bearers* (portadoras), a pesar de que las mismas existen ya que se establece correctamente la conexión en el plano de usuario. Una vez conectado el UE, este puede conectarse a la red IP externa a la que dé acceso el núcleo de red.

```
INFO:mme:mme_S1AP_ctx_mgmt_INITIAL_CONTEXT_SETUP_RESPONSE():294> Processing S1AP Initial Context Setup Response Message
INFO:mme:mme_S1AP_ctx_mgmt_INITIAL_CONTEXT_SETUP_RESPONSE():356> Successfully processed S1AP Initial Context Setup Response message
WARN:Platform:phoenix_mem_prealloc_realloc():603> MUplinkNASTransport: can't realloc 0x7f25832d22e8 because it is not the last allocation fr
om this pool
INFO:mme:mme_S1AP_nas_transport_UPLINK_NAS_TRANSPORT():262> Received S1AP Uplink NAS Transport message
NOTICE:mme:mme_NAS_sm_print_transaction():96>  ------------------
NOTICE:mme:mme_NAS_sm_print_transaction():98>  |    Session with IMSI [208930000000003]
NOTICE:mme:mme_NAS_sm_print_transaction():101>  |    Type: [01:ATTACH]
NOTICE:mme:mme_NAS_sm_print_transaction():103>  |    Current NAS Procedure: [19:NAS__ATTACH_COMPLETE]
NOTICE:mme:mme_NAS_sm_print_transaction():106>  ------------------
NOTICE:mme:mme_NAS_emm_specific_ATTACH_COMPLETE():806> Received NAS Attach Complete message
INFO:mme:mme_NAS_emm_specific_ATTACH_COMPLETE():821> Processing NAS Attach Complete message
WARN:Platform:str_dup_impl():110> empty source ph_str(trace: nas_msg_decode_hdr)
INFO:mme:mme_session_subscriber_state_update():144> Updating subscriber state from [0:UNREGISTERED] to [1:ACTIVE]
NOTICE:mme:mme_NAS_emm_specific_ATTACH_COMPLETE():837> Attachment successful - [208930000000003]
INFO:mme:mme_NAS_esm_network_init_ACTIVATE_DEFAULT_EPS_BEARER_CONTEXT_ACCEPT():240> Processing NAS ESM Activate Default EPS Bearer Context A
ccept message
INFO:gw:gw_gtp_s5s8_handle_modify_bearer_response():184> [208930000000003]: Handling Modify Bearer Response
WARN:gw:gw_send_packet_out():280> No packet_in data in binding, not sending packet_out!
INFO:gw:gw_gtp_s5s8_handle_modify_bearer_response():209> Forwarding Modify Bearer Response from SGW to MME
INFO:gtp_s11:gtp_s11_GTP_send_modify_bearer_response():1768> Sent GTP Modify Bearer Response message with TEID 00000001
NOTICE:mme:mme_GTP_sm_transport_handler():141> ** Received GTP-C message **
NOTICE:mme:mme_GTP_sm_print_transaction():104>  ------------------
NOTICE:mme:mme_GTP_sm_print_transaction():107>  |    Session with IMSI [208930000000003]
NOTICE:mme:mme_GTP_sm_print_transaction():109>  |    Type: [01:ATTACH]
NOTICE:mme:mme_GTP_sm_print_transaction():111>  |    Current GTP Procedure: [06:GTP__MODIFY_BEARER]
NOTICE:mme:mme_GTP_sm_print_transaction():114>  ------------------
INFO:mme:mme_GTP_tunnel_mgmt_MODIFY_BEARER_RESPONSE():1238> Received GTPC Modify Bearer Response message
INFO:mme:mme_GTP_tunnel_mgmt_MODIFY_BEARER_RESPONSE():1250> Processing GTPC Modify Bearer Response message
INFO:mme:mme_GTP_tunnel_mgmt_MODIFY_BEARER_RESPONSE():1364> Successfully processed GTPC Modify Bearer Response message
INFO:mme:mme_GTP_tunnel_mgmt_MODIFY_BEARER_RESPONSE():1386> * GTP-U: [ v1 213652889 3396329693 192.168.3.1 192.168.4.8]
INFO:mme:mme_GTP_sm_transport_handler():178> Successfully processed the GTP-C message
```

Figura 29: Señalización en el MME de *Open5GCore* durante la conexión del UE comercial

```
======================================== STATISTICS ========================================


               |   Current Status|  Added since last display|   Removed since last display |
Connected eNBs |          1      |             0              |             0
Attached UEs   |          1      |             1              |             0
Connected UEs  |          1      |             1              |             0
Default Bearers|          0      |             0              |             0
S1-U Bearers   |          0      |             0              |             0


======================================== STATISTICS ========================================
```

Figura 30: Registro de eNB y UE conectados al EPC de OAI

7.3. Evaluación de los núcleos de red LTE con *Landslide*

En esta sección se comentan los resultados de las pruebas realizadas al núcleo de red de OAI y a la *Release* 3 de *Open5GCore* para comparar su rendimiento. Cabe destacar que el acceso radio emulado por *Landslide* es ideal y no tiene errores, por lo que las diferencias se deben exclusivamente a los núcleos de red.

Por otro lado, el sistema presenta los resultados utilizando *slots* (intervalos de tiempo) de 15 segundos. De esta manera, además de obtener una media de la variable que se esté midiendo, pueden comprobarse las prestaciones del sistema examinado a lo largo de la prueba. Esta forma de operar también se tiene en cuenta a la hora de diseñar las pruebas.

Prueba de capacidad o establecimiento de sesión

El objetivo de esta prueba es determinar la capacidad del núcleo de red para gestionar peticiones de inicio de sesión. Se pretende averiguar el tiempo medio que transcurre desde que un UE envía la petición hasta que se establece el plano de usuario y puede comenzar a utilizar los recursos de la red.

Para la prueba se definen cuatro casos: la conexión de 50 y 100 usuarios en 45 y 90 segundos (3 y 6 intervalos). Se modifica la tasa de activación convenientemente para garantizar que todos los usuarios pueden conectarse durante el tiempo que dura la prueba

y que las peticiones de inicio de sesión se reparten uniformemente entre los intervalos.

La Figura 31 muestra una gráfica con las prestaciones del EPC de OAI. Las curvas roja y azul corresponden a la conexión de 50 UE, mientras que la negra y la amarilla son el resultado de la conexión de 100 UE. Puede identificarse un incremento en el tiempo medio de inicio de sesión de los usuarios en cada intervalo, de modo que con una base de usuarios conectados, el núcleo de red gestiona con más lentitud las nuevas peticiones.

Asimismo, la pendiente de las curvas es mayor cuando la conexión de los usuarios se realiza en 45 segundos. Esto denota que el núcleo de red experimenta mayores complicaciones si tiene que gestionar un número determinado de peticiones de inicio de sesión en un lapso de tiempo más corto.

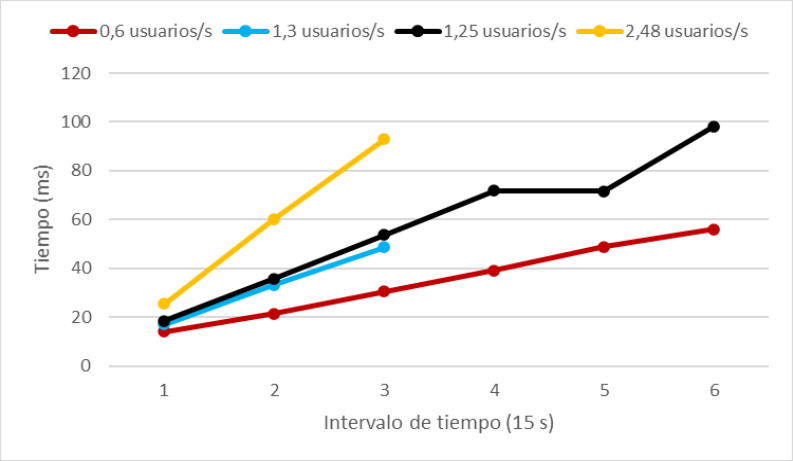

Figura 31: Tiempo medio de inicio de sesión por intervalo del EPC de OAI

Los resultados del EPC de *Open5GCore* pueden verse en la Figura 32. Las curvas roja y negra se refieren a la conexión de 50 UE, mientras que la azul y la amarilla corresponden a la conexión de 100 UE.

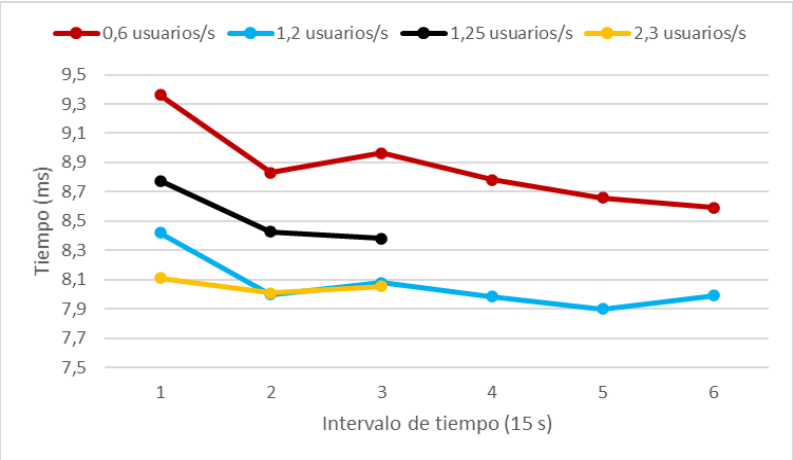

Figura 32: Tiempo medio de inicio de sesión por intervalo *Open5GCore*

En primer lugar, se observa una tendencia opuesta a la del caso anterior, puesto que el tiempo medio de inicio de sesión en el último intervalo de la prueba es inferior al tiempo registrado al principio de la misma. Además, puede considerarse la variable prácticamente estable, dado que experimenta un incremento de apenas 1 milisegundo al aumentar el número de usuarios que se conectan o la duración de la prueba. Finalmente, debe destacarse la diferencia en la magnitud del tiempo medio de inicio de sesión, mucho mayor en el núcleo de red de OAI.

En el marco de esta prueba también se ha comprobado el comportamiento de los núcleos de red con el número máximo de UE admitidos según sus configuraciones por defecto (rango de direcciones IP asignables a los UE disponibles). Son 126 usuarios en el caso de OAI y 253 en el caso de *Open5GCore*.

En ambos casos se mantiene la tendencia de las figuras anteriores, por lo que se intenta forzar el crecimiento en el tiempo medio de inicio de sesión del segundo núcleo de red. Se modifica la configuración definiendo una subred con muchas direcciones IP disponibles, logrando conectar alrededor de 1500 usuarios sin que varíe la gráfica.

Finalmente, también se prueba en este escenario la conexión de varios eNB a los núcleos de red para ver el número de asociaciones que admite el MME de cada uno. Se establece en 2 para el EPC de OAI y 10 para el de *Open5GCore*, aumentando en ambos casos el tiempo medio de inicio de sesión si cada usuario está conectado a un eNB distinto.

Prueba de carga de sesión

Mediante esta prueba se consigue determinar la capacidad del núcleo de red para procesar una sucesión de activaciones y desactivaciones de usuarios en un tiempo prolongado. Se diseña la prueba de forma que se alterna la conexión y la desconexión de todos los UE en cada intervalo.

Se establecen dos casos, con 25 y 100 usuarios, y una duración de 210 segundos (14 intervalos), con la mitad de intervalos reservados para conexión y la otra mitad para desconexión. En cada intervalo de conexión se mide, por un lado, el impacto de la carga en el tiempo medio de inicio de sesión. Por el otro, se comprueba el número de usuarios que realmente consiguen establecer una sesión con el núcleo de red y disponer del plano de usuario.

La Figura 33 muestra la gráfica del tiempo medio de inicio de sesión en el EPC de OAI de un UE en cada uno de los intervalos de conexión. Puede observarse que tanto si se realiza la prueba con 25 usuarios como con 100 el tiempo medio de inicio de sesión en cada intervalo aumenta. La progresión en el caso de conectar 100 UE es más irregular y mucho mayor en magnitud (segundos frente a milisegundos).

En cuanto a los usuarios que logran establecer sesión en cada intervalo de conexión, los resultados se presentan en la Tabla 13. En la prueba con 25 usuarios, se conectan en cada intervalo 22. No obstante, se producen los 175 inicios de sesión esperados a lo largo de la prueba, ya que los inicios de sesión restantes se producen en los intervalos de desconexión. En el caso de conectar 100 UE, el rendimiento del núcleo de red es notablemente peor y no se establecen los 700 inicios de sesión esperados.

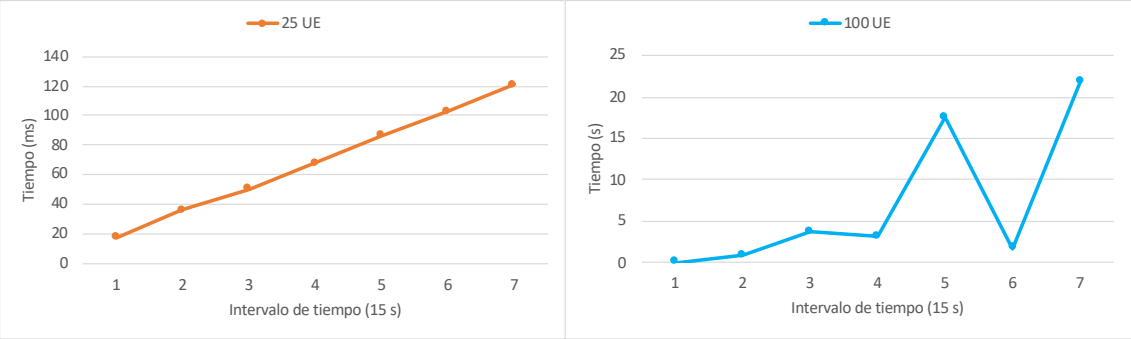

Figura 33: Tiempo medio de inicio de sesión en la prueba de carga del EPC de OAI

	1	2	3	4	5	6	7
Prueba con 25 UE	23	22	22	22	22	22	22
Prueba con 100 UE	97	80	43	20	53	2	24

Tabla 13: UE iniciando sesión en cada intervalo de la prueba de carga (OAI EPC)

Por su parte, las prestaciones del EPC de *Open5GCore* se pueden observar en la gráfica de la Figura 34. Como era de esperar tras la prueba de capacidad, los tiempos medios de inicio de sesión son notablemente inferiores a los del núcleo de red de OAI, y apenas varían a lo largo del examen. Por tanto, este núcleo de red completa con solvencia la prueba. Respecto a los inicios de sesión que se producen en cada intervalo (Tabla 14), ocurre lo mismo que en el caso anterior. Sin embargo, a diferencia de aquel, en la prueba con 100 UE consiguen establecerse los 700 inicios de sesión esperados.

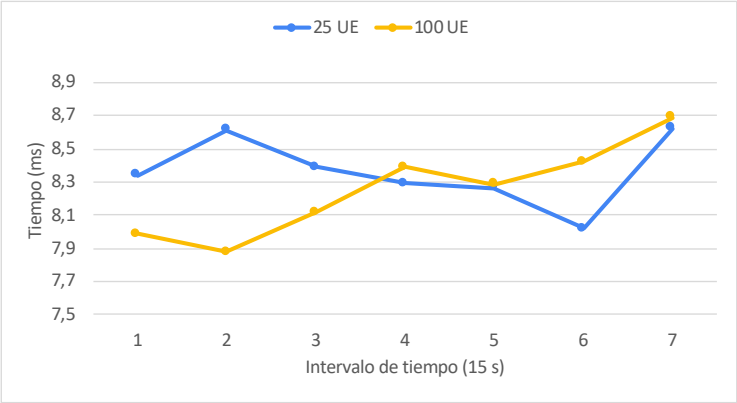

Figura 34: Tiempo medio de inicio de sesión en la prueba de carga de *Open5GCore*

	1	2	3	4	5	6	7
Prueba con 25 UE	20	20	20	20	20	19	19
Prueba con 100 UE	83	82	81	80	80	80	79

Tabla 14: UE iniciando sesión en cada intervalo de la prueba de carga (*Open5GCore*)

Capítulo 8

Conclusiones y propuestas de trabajo futuro

A lo largo de este Trabajo Final de Grado se ha completado la implementación de núcleos de red LTE y 5G en el marco del despliegue Valencia Campus 5G llevado a cabo por la Universitat Politècnica de València.

El trabajo ha consistido, en primer lugar, en una revisión profunda del estado del arte, así como en la recopilación de las funciones y entidades que debían desplegarse.

En segundo lugar, ha sido necesario un proceso de aprendizaje muy intenso respecto al manejo de las tecnologías de virtualización, puesto que la solución de núcleos de red 5G está intrínsecamente relacionada con la operación de máquinas virtuales y contenedores.

A continuación, se desplegaron los núcleos de red de las plataformas *OpenAirInterface* y *Open5GCore* y se interconectaron las entidades radio de las tecnologías 4G y 5G necesarias. Este proceso supuso un enorme esfuerzo, sobre todo porque la parte radio se heredó de la solución OAI, en fase de desarrollo para la implementación de la tecnología radio 5G. Por otro lado, *Open5GCore* no forma parte de dicha solución y fue necesario realizar una configuración alternativa a la definida por defecto para lograr tener una red móvil plenamente operativa.

Finalmente, se realizó un análisis en profundidad de las prestaciones de ambos núcleos de red, comparando su fiabilidad, carga máxima y velocidad. Para ello se hizo uso de las capacidades de testeo de *Landslide*, herramienta muy completa y comercial de validación de núcleos de red compatible con LTE y 5G.

Las conclusiones alcanzadas tras este proceso son múltiples. En primer lugar, es importante destacar que las soluciones de núcleos de red de código abierto están adquiriendo muchísima importancia en los últimos meses. Desde que se inició el trabajo de este TFG hasta el momento de la escritura de la memoria, se ha observado cómo la comunidad de empresas y entidades participantes se ha incrementado exponencialmente. Esto augura un futuro exitoso para la línea de trabajo desarrollada en este proyecto, pero también complica mucho la gestión de las distintas versiones y la integración de las mismas. Sin embargo, se ha demostrado que las soluciones de código abierto LTE y 5G funcionan, y ello es una muy buena noticia para la sociedad puesto que supondrá una importante reducción de

costes para los operadores y, por tanto, de los precios para los consumidores. Además, disponer de un código abierto también permite comprobar las condiciones de seguridad de la solución implementada, lo que acaba suponiendo una mayor tranquilidad para usuarios y Estados, especialmente preocupados en el contexto geopolítico actual.

Respecto a la comparativa entre *Open5GCore* y el EPC de OAI, todas las pruebas realizadas corroboran que la solución de OAI es muy poco robusta y no es una opción para despliegues reales de los que se espere una funcionalidad estable. Por su parte, *Open5GCore* soporta más de 100 usuarios por intervalo de acceso y un total de más de 1500 usuarios conectados, más que suficientes para muchas de las aplicaciones de despliegue de redes privadas 5G que se están barajando en la actualidad. Puede concluirse, por tanto, que *Open5GCore* es la mejor opción para el despliegue del Campus 5G.

Como líneas futuras de trabajo, se plantean las siguientes:

- Completar las pruebas de prestaciones una vez esté disponible la actualización de la componente radio de OAI que permita establecer portadoras de datos sobre la interfaz X2 para configurar un escenario 5G NSA.
- Realizar la evolución hacia una red 5G SA, desplegando la *Release* 5 de *Open5GCore* con un acceso radio basado en NR.
- Desplegar nuevas funciones de red programando los protocolos de interconexión definidos por el estándar. Se han identificado algunas funciones muy relevantes que no están disponibles todavía, sobre todo las relativas a la gestión de estadísticas y mecanismos de inteligencia artificial para la operación eficiente de la red.
- Extender la parte radio del Campus 5G incluyendo, por ejemplo, la solución de *Amarisoft* disponible en el MCG y solventando los problemas de estabilidad de *Open5GCore* al conectar dos accesos radio con distintas versiones de señalización.
- Investigar alternativas para la creación de *Network Slices* que permitan aislar distintas redes dentro del mismo despliegue. Estudiar sus prestaciones.

Bibliografía

[1] M. Lynne Markus y Daniel Robey. «Information Technology and Organizational Change: Causal Structure in Theory and Research». En: *Management Science* 34.5 (1988), págs. 583-598. ISSN: 0025-1909. DOI: 10.1287/mnsc.34.5.583. URL: https://www.jstor.org/stable/2632080.

[2] José F. Monserrat y col. *3GPP LTE-Advanced y su evolución hacia la 5G móvil*. Barcelona: Marcombo, 2017. ISBN: 9788426724472.

[3] Vasco Pereira y Tiago Sousa. «Evolution of Mobile Communications: from 1G to 4G». En: *Department of Informatics Engineering of the University of Coimbra, Portugal* (2004).

[4] ITU. *5G - Fifth generation of mobile technologies*. 2019. URL: https://www.itu.int/en/mediacentre/backgrounders/Pages/5G-fifth-generation-of-mobile-technologies.aspx (visitado 16-08-2020).

[5] Ms. Anju Uttam Gawas. «An Overview on Evolution of Mobile Wireless Communication Networks: 1G-6G». En: *International Journal on Recent and Innovation Trends in Computing and Communication* 3.5 (2015), págs. 3130-3133. ISSN: 2321-8169.

[6] Mamta Agiwal, Abhishek Roy y Navrati Saxena. «Next generation 5G wireless networks: A comprehensive survey». En: *IEEE Communications Surveys and Tutorials* 18.3 (2016), págs. 1617-1655. ISSN: 1553877X. DOI: 10.1109/COMST.2016.2532458.

[7] Ericsson. *Ericsson Mobility Report*. Inf. téc. Junio. 2020. URL: https://www.ericsson.com/en/mobility-report/reports/june-2020.

[8] Xiaofeng Tao y col. «Recent advances and future challenges for mobile network virtualization». En: *Science China Information Sciences* 60.4: 040301 (2017), págs. 1-12. ISSN: 1674733X. DOI: 10.1007/s11432-017-9045-1.

[9] Hsin Hung Cho y col. «Integration of SDR and SDN for 5G». En: *IEEE Access* 2 (2014), págs. 1196-1204. ISSN: 21693536. DOI: 10.1109/ACCESS.2014.2357435.

[10] Magnus Olsson y col. *SAE and the Evolved Packet Core: Driving The Mobile Broadband Revolution*. Academic Press, 2009. ISBN: 9780123748263.

[11] LTE Encyclopedia. *LTE Network Infrastructure and Elements*. URL: https://sites.google.com/site/lteencyclopedia/lte-network-infrastructure-and-elements (visitado 10-09-2020).

[12] Francisco García Espigares. «Prototipo de una estación base 4G usando Open Air Interface». Tesis doct. Universidad de Granada, 2017.

[13] Peter Schmitt, Bruno Landais y Frank Yong Yang. *Control and User Plane Separation of EPC nodes (CUPS)*. 2017. URL: https://www.3gpp.org/news-events/1882-cups (visitado 10-09-2020).

[14] Stefan Rommer y col. *5G Core Networks: Powering Digitalization*. Academic Press, 2020. ISBN: 9780081030097.

[15] GSMA. «5G Implementation Guidelines: NSA Option 3». En: February (2020). URL: https://www.gsma.com/futurenetworks/wp-content/uploads/2019/03/5G-Implementation-Guidelines-NSA-Option-3-v2.1.pdf.

[16] Margaret Chiosi y col. *Network Functions Virtualisation - Introductory White Paper*. Inf. téc. 2012. URL: http://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[17] Jim Doherty. *SDN and NFV Simplified: A visual guide to understand Software Defined Networks and Network Function Virtualization*. Addison-Wesley Professional, 2016. ISBN: 9780134306407.

[18] Van-Giang Nguyen y col. «SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey». En: *IEEE Communications Surveys and Tutorials* 19.3 (2017), págs. 1567-1602. ISSN: 1553877X. DOI: 10.1109/COMST.2017.2690823.

[19] ETSI ISG NFV. «Network Functions Virtualisation (NFV); Use Cases». En: (2013). URL: https://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf.

[20] Faqir Zarrar Yousaf y col. «NFV and SDN-Key technology enablers for 5G networks». En: *IEEE Journal on Selected Areas in Communications* 35.11 (2017), págs. 2468-2478. ISSN: 07338716. DOI: 10.1109/JSAC.2017.2760418.

[21] Mosaic5G. *Home*. URL: http://mosaic-5g.io/ (visitado 08-09-2020).

[22] 5G-PPP. *About us*. URL: https://5g-ppp.eu/ (visitado 08-09-2020).

[23] Taesang Choi y col. «Agile Management and Interoperability Testing of SDN/NFV-Enriched 5G Core Networks:» en: *ETRI Journal* 40.1 (2018), págs. 72-88. ISSN: 22337326. DOI: 10.4218/etrij.2017-0236. URL: http://doi.wiley.com/10.4218/etrij.2017-0236.

[24] OpenAirInterface. *OpenAirSoftwareSupport | GitHub*. 2020. URL: https://github.com/OPENAIRINTERFACE/openair-cn/wiki/OpenAirSoftwareSupport (visitado 05-09-2020).

[25] Super Micro Computer Inc. *1029P-WTRT | 1U | SuperServers | Products*. URL: https://www.supermicro.com/en/products/system/1U/1029/SYS-1029P-WTRT.cfm (visitado 05-09-2020).

[26] Ettus Research. *USRP B210 USB Software Defined Radio (SDR)*. 2019. URL: https://www.ettus.com/all-products/ub210-kit/ (visitado 05-09-2020).

[27] Open Cells. *SIM cards – 4G and 5G reference software*. URL: https://open-cells.com/index.php/sim-cards/ (visitado 05-09-2020).

[28] OpenAirInterface. *TESTING GNB W COTS UE · openairinterface5G · GitLab*. 2020. URL: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/TESTING_GNB_W_COTS_UE.md (visitado 10-09-2020).

[29] Spirent Communications. *Landslide Core Network Testing*. URL: https://www.spirent.com/products/core-network-test-5g-lte-ims-wifi-diameter-landslide (visitado 09-09-2020).

[30] Spirent Communications. *Spirent Landslide™ Virtual Functional and performance testing for Mobile, Wi-Fi, IMS and Diameter networks*. Inf. téc. 2018. URL: https://tinyurl.com/y4ax5j82.

[31] Spirent Communications. *Spirent Landslide™ C100-M4*. Inf. téc. 2019. URL: https://tinyurl.com/y2hfgcwz.