



UNIVERSITAT POLITÈCNICA DE CATALUNYA

Escola Politècnica Superior d'Enginyeria de Manresa

Segmentació i monitoratge d'una xarxa LTE de codi obert

13 de gener de 2021

treball de fi de grau que presenta

DIEGO CARREÑO ASUETA

en compliment dels requisits per assolir el

GRAU D'ENGINYERIA EN SISTEMES TIC

Direcció: Ilker Seyfettin Demirkol

Aquesta obra està subjecta a una llicència Attribution-NonCommercial-ShareAlike 3.0 Spain de Creative Commons. Per veure'n una còpia, visiteu <https://creativecommons.org/licenses/by-nc-sa/3.0/es/deed.ca> o envieu una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

A Joan Antoni, gràcies per tot. Ha arribat el fi de l'aventura.

Agraïments

Agraeixo a tots els meus professors que he tingut per tenir la bondat i paciència d'ensenyar-me tot el que he après fins ara. En especial a tots els docents de l'EPSEM i al meu tutor Ilker Demirkol que sense el seu suport i coneixements aquest projecte no s'hagués pogut dur a terme.

Agreixo a tots els meus familiars i amics que han estat al meu costat donant-me suport i ànim sempre que ho he necessitat. Moltes gràcies a tothom.

Resum

En aquest treball es realitza un prototip de xarxa LTE (*Long Term Evolution* o Evolució a Llarg Termini) amb l'objectiu de monitorar-la i segmentar els seus recursos en diferents *slices* (o segments). Per tal de dur-ho a terme es segueixen els estàndards oficials que presenta el 3GPP (*Third Generation Partnership Project* o Projecte d'Associació de Tercera Generació).

El prototip de xarxa emprà el programari de *software* de lliure accés a través de la plataforma de codi obert OAI (*Open-Air-Interface*) i Mosaic5G. La configuració dels principals elements de la xarxa es realitza sobre una mateixa estació de treball i l'UE (*User Equipment* o Equip de l'Usuari) és un *dongle* que té inserida una targeta SIM (*Subscriber Identity Module* o Mòdul d'Identificació del Subscriptor). Aquest *dongle* fa de *hotspot* (o punt calent) a una *Raspberry* que és l'element que s'usa al prototip per a dur a terme la verificació del projecte.

Abstract

In this project a LTE (Long Term Evolution) network has been developed in order to monitor it and its consequent segmentation. Thereinafter, the official standards established by the 3GPP (Third Generation Partnership Project) are taken into consideration.

The network's prototype is based on the software proportioned by open-code platforms named OAI (Open-Air-Interface) and Mosaic5G, and its main constituents configuration are developed under the same working machine. What is more, the UE (User Equipment) is a *dongle* which has a SIM (Subscriber Identity Module), and acts as a hotspot to a *Raspberry*, the element used in the prototype to verify the project.

Índex

Resum	i
Abstract	i
I. Memòria	1
1. Introducció	3
1.1. Context	3
1.2. Motivació	5
1.3. Objectius	6
1.4. Resultats	6
1.5. Organització de la memòria	7
2. Marc teòric	9
2.1. Introducció	9
2.2. LTE (<i>Long Term Evolution</i>)	9
2.2.1. Evolució des de les xarxes 2G i 3G	9
2.2.2. Introducció a la tecnologia LTE	11
2.2.3. Arquitectura de xarxa: eNB	13
2.2.4. Arquitectura de xarxa: EPC	14
2.2.5. Protocols de xarxa	16
2.3. SDR (<i>Software Defined Radio</i>)	18
2.3.1. Descripció general	18
2.3.2. USRP (<i>Universal Software Radio Peripheral</i>)	19
2.4. Programari de codi obert	19
2.4.1. <i>Open-Air-Interface</i> (OAI)	19
2.4.2. Mosaic5G	20
2.4.3. <i>Open Cells</i>	20
2.5. MySQL i phpMyAdmin	21
2.6. Eines de verificació i monitoratge de la xarxa	21
2.6.1. <i>iPerf3</i>	21
2.6.2. <i>Wireshark</i>	21
2.6.3. <i>FlexRAN</i>	21
3. Especificacions i restriccions del muntatge	23
3.1. Especificacions	23
3.2. Restriccions	24

4. Descripció del muntatge	25
4.1. Descripció general	25
4.2. Configuració inicial de l'escenari	26
4.2.1. Configuració inicial de la màquina eNB	26
4.2.2. Configuració inicial de la màquina EPC	27
4.3. Instal·lació de Mosaic5G	28
4.4. Descripció del muntatge	29
4.4.1. Configuració de l'EPC	29
4.4.2. Configuració de l'eNB	34
4.4.3. Configuració i gestió d'usuaris de l'HSS	36
4.4.4. Configuració del dongle	39
4.4.5. Configuració del <i>FlexRAN</i>	41
4.4.6. Configuració del <i>Slicing</i>	41
5. Resultats	43
5.1. Captures de xarxa	43
5.1.1. Plànol de control	43
5.1.2. Plànol d'usuari	44
5.2. Proves de rendiment, monitoratge i segmentació	45
5.2.1. Funcionament de <i>FlexRAN</i>	45
5.2.2. Proves de rendiment	48
6. Conclusions	59
7. Treball futur	61
Bibliografia	63
II. Annexos	67
A. Configuració inicial de la màquina eNB	69
A.1. Instal·lació d'un <i>Kernel lowlatency</i>	69
A.2. Control de la gestió de potència	70
A.3. Configuració del <i>hostname</i>	73
A.4. Configuració d'una IP estàtica	73
B. Configuració inicial de la màquina EPC	75
B.1. Instal·lació de <i>VMWare</i> i creació de la màquina virtual	75
B.2. Instal·lació d'un <i>kernel</i> amb suport de GTP	75
B.3. Configuració del <i>hostname</i>	76
B.4. Configuració d'una IP estàtica	77
C. Arxius de configuració del projecte	79
C.1. Arxiu de configuració de l'eNB	80

C.2. Arxius de configuració de <i>FlexRAN</i>	88
C.2.1. Creació de <i>slices</i>	88
C.2.2. Assignació de <i>slices</i>	89
C.3. Arxius de configuració de l'EPC	90
C.3.1. Arxius de configuració de l'HSS	90
C.3.2. Arxius de configuració de l'MME	93
C.3.3. Arxiu de configuració de l'SPGW	102

Part I.

Memòria

1. Introducció

1.1. Context

Des del moment en què en Martin Cooper va crear el que es considera com el primer mòbil de la història l'any 1973 fins avui dia, les xarxes mòbils han experimentat un augment tant en el seu ús com en les seves funcionalitats. En l'actualitat les xarxes de comunicació mòbil s'han convertit en una part fonamental en les xarxes de telecomunicacions.

L'augment del seu ús té relació amb l'alta gamma de funcionalitat que aquesta tecnologia pot oferir. Presten serveis de mobilitat que cada vegada són més exigents per la continuada demanda dels usuaris i, a més, representen una branca d'investigació molt important que engloba molts àmbits de l'enginyeria.

A Espanya des de 2007 fins a 2019 les línies telefòniques de mòbil han crescut notablement, especialment les de postpagament, que durant aquest període han augmentat des dels 27.6 milions de subscripcions fins a més de 44.6 milions tal com es pot apreciar a la figura 1.1, extreta de la pàgina web statista:

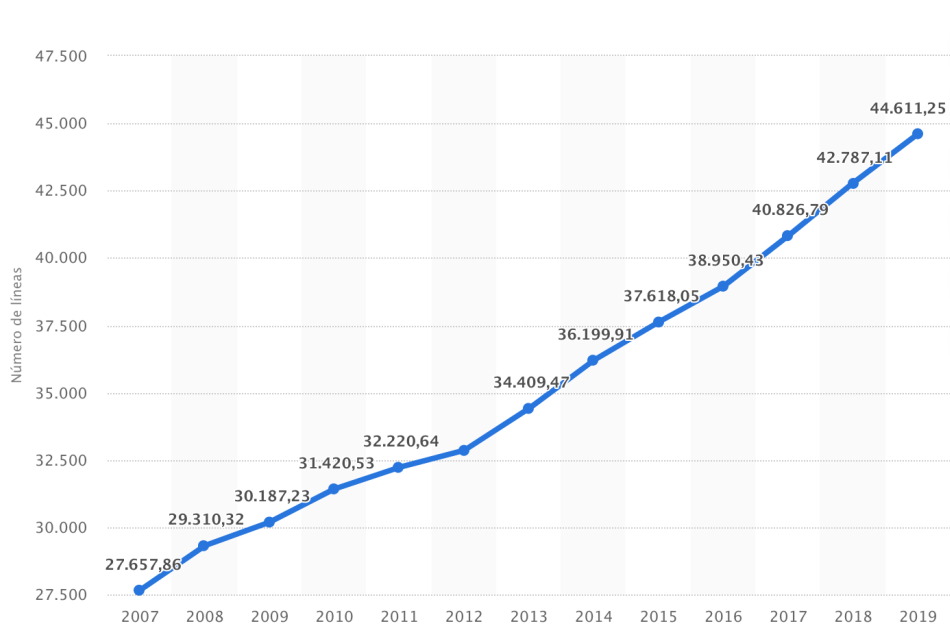


Figura 1.1.: Nombre de contractes de telefonia mòbil de postpagament [7].

A la resta del món aquest avanç mencionat es pot veure reflectit a través de dades estadístiques proporcionades per la *Global Mobile Suppliers Assocation* (GSMA) que, entre altres dades, afirma que a desembre de 2020 existeixen més de 5 mil milions de subscripcions de telefonia mòbil. I la tendència és continuar creixent, ja que per l'any 2025 s'estima que les subscripcions pugin fins a 5.795 milions [11].

Un altre aspecte que ha afavorit el creixement d'aquest tipus de xarxa és l'aparició i desenvolupament de tecnologies que l'han anat millorant. Des de la 1G (primera generació) on es presentava un funcionament totalment analògic, millorada per la 2G (segona generació) que es va centrar en el pas d'analògic a digital, que va evolucionar fins a la 3G (tercera generació) on ja es podia mantenir una connexió a internet fins a arribar a la 4G (quarta generació) on el seu estàndard està basat totalment en la IP (*Internet Protocol* o Protocol d'Internet). Aquestes tecnologies serveixen de base per la pròxima generació, 5G (cinquena generació), que cada vegada està més present en el dia a dia.

La contínua millora que hi ha entre generació i generació fa que cada vegada les tecnologies més antigues estiguin cada cop més en desús. La GSMA preveig que cap a l'any 2025 1.8 mil milions de connexions mòbils seran via 5G, això suposa un 20% de totes les connexions mòbils mundials. A la figura 1.2 es pot apreciar en quin percentatge les tecnologies seran utilitzades l'any 2025 en comparació l'any 2019:

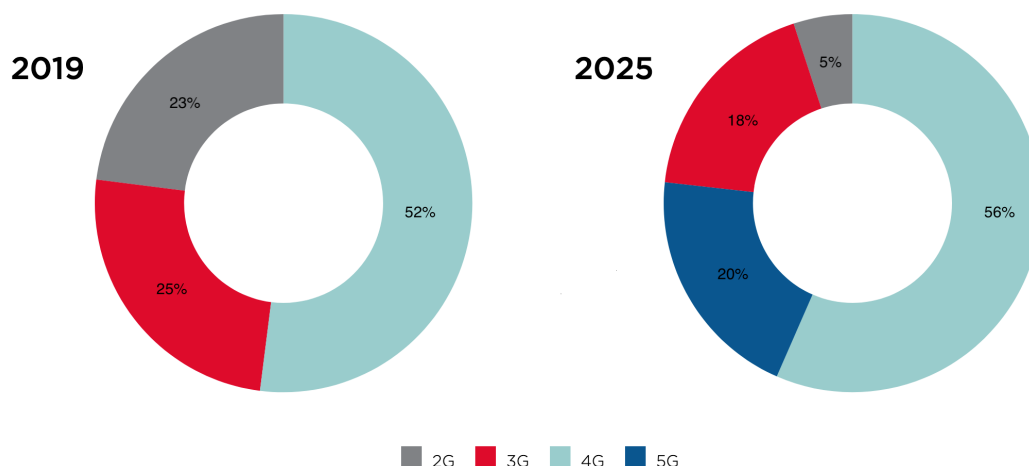


Figura 1.2.: Comparativa de la utilització de les diferents tecnologies [11].

Les dades indiquen el continu creixement de les noves tecnologies en contra part amb les antigues. Encara que la quinta generació comenci a ser usada, la xarxa que més s'usa actualment i, fins a almenys l'any 2025, és la xarxa 4G-LTE (*Long Term Definition* o Evolució a Llarg Termini) [11]. Tal com es pot veure a la figura 1.3, les subscripcions a aquesta xarxa no han parat de pujar des de 2011 fins avui dia i, arribaran al seu

màxim l'any 2022, on la cinquena generació començarà a tenir més protagonisme i les subscripcions a la xarxa LTE disminuiran.

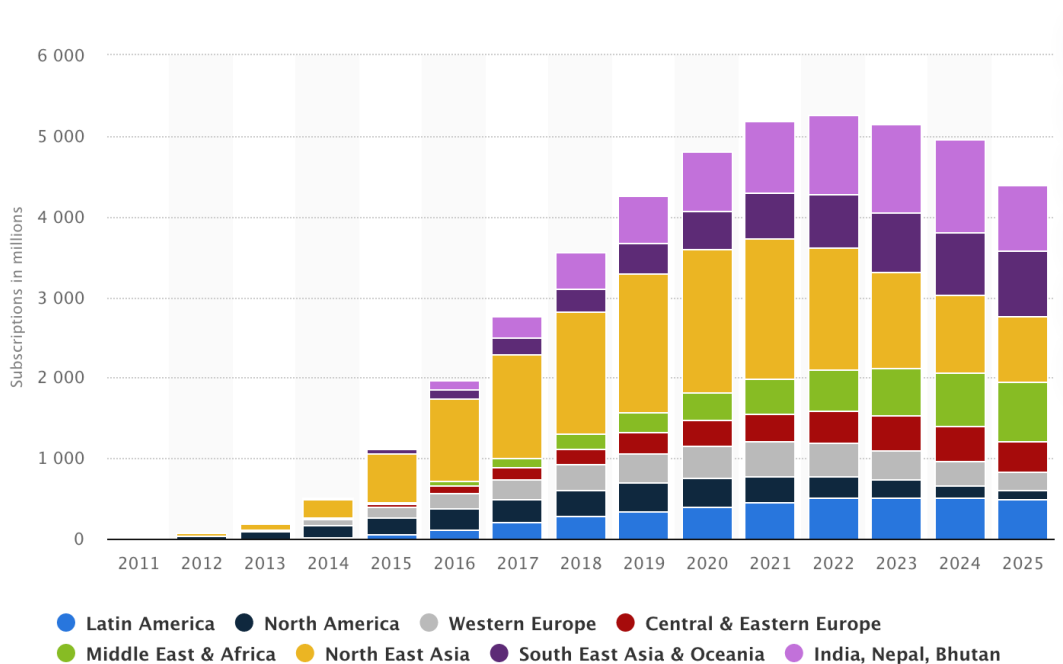


Figura 1.3.: Subscripcions a una xarxa LTE a tot el món.

1.2. Motivació

Com s'ha vist a la secció anterior, les xarxes mòbils estan en continu creixement i millora. Gràcies al seu desenvolupament poden proporcionar solucions per satisfer la gran demanda de comunicació que el públic presenta on la protagonista és la xarxa 4G i, en un futur, la 5G. Per això es pot afirmar tant que la xarxa de quarta generació com la cinquena tenen una importància molt gran en les TIC (Tecnologies de la Informació i la Comunicació). És important continuar amb la investigació i el progrés d'aquestes tecnologies per tant de complir tots els requisits que els usuaris demanen (altes velocitats, baix temps de resposta, alta disponibilitat, etcètera).

Per aquest motiu és necessari que els titulats puguin posar en pràctica els coneixements teòrics d'aquest tipus de xarxes, ja que, com s'ha vist en les estadístiques, la demanda pujarà i, per tant, seran imprescindible l'augment de professionals en aquest camp. Dins aquest context tenir la possibilitat de crear un prototip amb programari de codi obert per a fins didàctics que dona la possibilitat de monitorar la xarxa i dividir els seus recursos en diferents *slices* (o segments) tal com es realitza als casos reals i, així, poder posar en pràctica el que s'ha après resulta especialment interessant.

1.3. Objectius

L'objectiu principal que es vol aconseguir és el desplegament d'un prototip d'una xarxa LTE per a la seva posterior segmentació i monitoratge. Aquesta xarxa es du a terme mitjançant la utilització de sistemes SDR (*Software Defined Radio* o Ràdio Definida per *Software*) i emprant les llibreries de *software* de codi obert OAI (*Open-Air-Interface*) i Mosaic5G.

D'acord amb l'objectiu principal s'extreuen els següents objectius parcials:

- Complir les especificacions de *hardware* i de *software* que OAI imposa per la realització dels seus tutorials.
- Configurar els diferents elements de la xarxa LTE: eNB (*Evolved NodeB* o Evolucionat Node B) i l'EPC (*Evolved Packet Core* o Nucli de Paquet Evolucionat).
- Configurar l'UE (*User Equipment* o Equip de l'Usuari) que en aquest projecte és un *dongle* que fa de *hotspot* (o punt calent) a una *Raspberry*.
- Verificar els objectius anteriors amb la utilització de les eines *iPerf3* i *Wireshark* per a examinar el correcte funcionament de la xarxa.
- Utilitzar l'instrument de monitoratge *FlexRAN* per tal de veure el comportament de la xarxa LTE.
- Dividir els recursos de la xarxa en diferents *slices* (o segments) i vincular l'UE a una d'elles utilitzant *FlexRAN*.

1.4. Resultats

El resultat és un prototip de xarxa LTE funcional capaç de donar connexió a internet a una *Raspberry* a través d'un *dongle* amb una targeta SIM. A més, s'ha aconseguit una correcta configuració i preparació dels diferents components de la xarxa així com la comprovació de la comunicació entre ells i internet mitjançant *Wireshark*. Amb *iPerf3* s'ha comprovat el rendiment del prototip que ha sigut capaç de donar velocitats de transferències de pujada de 7.5 MBits (Mega Bits) i de baixada de fins als 15.7 MBits.

Amb *FlexRAN* s'ha pogut observar com canvia l'estat de la xarxa en tot moment. Podent veure quan els diferents components es connecten entre si i, així, poder veure les especificacions de tots ells. També mostra el moment en què un UE es connecta a la nostra xarxa i, d'aquesta forma, veure les seves característiques.

Finalment, s'ha dut a terme la segmentació de la xarxa i s'ha definit els recursos per cada *slice* per la posterior associació de l'UE.

1.5. Organització de la memòria

La memòria consta de diversos capítols on cadascun d'ells té la intenció de proporcionar les explicacions necessàries per a l'enteniment del treball realitzat.

El primer capítol explica la situació de les xarxes mòbils, el seu creixement i la seva importància. Es relaciona aquesta tecnologia amb l'àmbit TIC i els seus estudiants, es declaren els objectius del projecte i, per últim, s'exposa el que s'ha aconseguit.

Al segon capítol es troba una explicació sobre la teoria necessària per a l'enteniment de disseny de la xarxa LTE. Es comença amb una breu introducció sobre les tecnologies 2G i 3G i s'inicien conceptes com eNB, EPC, MME (*Mobility Management Entity* o Entitat de Gestió de la Movilitat) o HSS (*Home Subscriber Server* o Servidor de Subscriptors). A més d'una explicació sobre què és i per a què serveix un SDR i de les eines que s'utilitzen al projecte.

Al tercer capítol es descriuen les especificacions i restriccions que té la xarxa.

El quart capítol explica la solució proposada on s'inclou l'arquitectura del sistema a desplegar i les tasques a realitzar per la creació del sistema.

Al cinquè capítol es realitzen les proves de funcionament de la xarxa amb les eines *iPerf3* i *Wireshark*, així com el seu monitoratge amb *FlexRAN* i les proves d'*slicing*.

Al sisè capítol es mostren les conclusions obtingudes en finalitzar el projecte i posterior anàlisi mitjançant les proves de funcionament i monitoratge.

Al setè capítol es presenten diverses vies per a continuar o millorar el projecte.

Finalment, als annexos, es descriuen els passos necessaris per a la preparació inicial de les màquines eNB i EPC i els arxius de configuració empleats pel disseny de la xarxa.

2. Marc teòric

2.1. Introducció

En aquest apartat es descriu el marc tecnològic on es situa el projecte. Es comença amb una explicació de com han evolucionat les xarxes mòbils, descrivint les tecnologies 2G i 3G, fins a arribar a la quarta generació. A continuació, s'explica la xarxa LTE: la seva arquitectura i característiques més importants. Després, es veu una descripció del concepte de SDR i la seva importància pel desenvolupament d'aplicacions que facin ús de la tecnòloga de ràdio. Posteriorment es descriuen les llibreries de codi obert emprades per la realització del prototip: *Open-Air-Interface* i *Mosaic5G* i, a més, s'exposaran les aplicacions per gestió de dades: *MySQL* i *phpMyAdmin*. Finalment, s'introduiran les eines de monitoratge usades per la verificació del funcionament de la xarxa: *iPerf3*, *Wireshark* i *FlexRAN*.

2.2. LTE (*Long Term Evolution*)

2.2.1. Evolució des de les xarxes 2G i 3G

Segona generació: 2G

A la segona generació l'estàndard dominant és GSM (*Global System for Mobile Communications* o Sistema Global per les Comunicacions Mòbils). Aquesta generació marca el pas de la telefonia analògica a la digital que mitjançant una sèrie de protocols va millorar del maneig de crides, va permetre més enllaços simultanis a la mateixa amplada de banda i la integració d'altres serveis addicionals al de la veu, com el Servei de Missatges Curts o SMS (*Short Message Service*).

Els estàndards varen ser implementats per diferents companyies. Aquest fet va donar origen al principal problema de la tecnologia: la incompatibilitat entre protocols. Per aquest motiu, en aquesta generació, el radi d'ús del dispositiu quedava limitat a l'àrea en què el seu operador li donés suport.

És a la segona generació quan es va definir l'ús de la targeta SIM (*Subscriber Identity Module* o Mòdul d'Identificació del Subscriptor). La idea és que per poder usar la xarxa s'ha de tenir una d'aquestes targetes, la qual, guarda la informació necessària de l'usuari per a poder connectar-se a les estacions base. Informació com: l'IMSI (*International Mobile Subscriber Identity* o Identitat Internacional del Subscriptor Mòbil), el MCC (*Mobile Country Code* o Codi Mòbil de País) i el MNC (*Mobile Network Code* o Codi Mòbil de Xarxa).

La xarxa d'accés de radi, denominada GERAN (GSM EDGE *Radio Access Network* o Xarxa d'Accés de Radi GSM i EDGE), està formada per: BTS (*Base Transceiver Station* o Estació Base Transceptora) que s'encarreguen de fer la connexió amb l'usuari i, les BSC (*Base Station Controller* o Controlador d'Estacions Base) que controlen les BTS.

Els estàndards més representatius de la generació són: GSM, GPRS (*General Packet Radio Service* o Servei General de Ràdio per Paquets) i EDGE (*Enhanced Data rates for GSM Evolution* o Tasses de Dades Millorades per l'evolució de GSM).

El primer d'ells (GSM) és el protocol més característic de la segona generació. Està basat en la commutació de circuits i en el TDMA (*Time Division Multiple Access* o Accés Múltiple per Divisió en els Temps) que consisteix en el fet que diversos usuaris puguin utilitzar un canal comú per a comunicar-se. Per exemple, a una trucada, a cada usuari se li assigna una ranura de temps per la transmissió de la seva veu. Amb aquesta tecnologia el circuit queda reservat fins que acabi la conversació.

L'evolució de GSM arriba amb el nom de GPRS. Aquesta tecnologia també està basada en TDMA amb la gran diferència de què treballa amb la commutació de paquets de control per saber si pot accedir al canal. Amb aquesta tècnica s'aconsegueix que encara que la comunicació entre dos usuaris no s'hagi acabat, si no s'estan enviant cap dada, les freqüències queden lliures per ser utilitzades per altres usuaris. D'aquesta forma es poden assignar més d'un canal a cada comunicació sense perill de saturar la xarxa.

El final de què es considera la segona generació arriba amb el protocol EDGE, l'evolució de GPRS. EDGE funciona exactament igual que GPRS i utilitza la mateixa tecnologia. El que la fa diferent és una nova capa física entre GPRS i HSCSD (*High-Speed Circuit Switched Data* o Conmutació de Circuits de Dades d'Alta Velocitat) i un nou esquema de modulació: 8-PSK (*Phase Shift Keying* o Modulació per Desplazament de Fase) que consisteix a fer variar la fase de la portadora entre un nombre determinat de valors discrets. Els avantatges principals d'EDGE sobre GPRS són la possibilitat d'augmentar el nombre d'usuaris a una operadora i la triplicació de la capacitat de transportació de dades.

Tercera generació: 3G

La tercera generació té com a objectiu l'augment de la velocitat de transmissió i recepció de dades. Per aquest motiu es va desenvolupar, entre altres, l'estàndard UMTS (*Universal Mobile Terrestrial System* o Sistema Universal Mòbil Terrestre) que usa la tecnologia d'accés WCDMA (*Wideband Code Division Multiple Access* o Accés Múltiple per Divisió de Codi de banda Ampla).

Aquesta nova tecnologia d'accés al medi és diferència de les anteriors en què el missatge que s'envia es multiplica bit per bit per un codi únic que l'identifica. Té dues formes de funcionament: TDD (*Time-Division Duplexing* o Dúplex per divisió de Temps) on les transmissions de pujada o baixada s'envien en la mateixa freqüència, però sincronitzades

en el temps i FDD (*Frequency Division Duplex* o Dúplex per Divisió de Freqüència) on les transmissions de pujada o baixada s'envien per bandes de freqüències amb una separació específica assignada per cada enllaç.

A UMTS s'emula una connexió amb commutació de circuits per a serveis en temps real (com una trucada) i una connexió amb commutació de paquets per a serveis de comunicació de dades. Quan un UE es connecta a la xarxa se li assigna una adreça IP i se li allibera de la IP quan es desconnecta del servei. Per tant, els serveis entrants continuen depenent del nucli del circuit commutat per a la cerca de persones.

A aquesta generació la xarxa d'accès de radi es denomina UTRAN (*Universal Terrestrial Radi Access Network* o Xarxa Universal d'Accés Ràdio Terrestre), la qual, està formada pel NodeB (abreujat NB) que és el component responsable de la transmissió i recepció radio entre el terminal UE i la xarxa, controlat pel RNC (*Ràdio Network Controller* o Controlador de Radi de la Xarxa).

Amb l'arribada del 3G les xarxes mòbils experimentaren una versatilitat de la qual abans no gaudien i milloraren les seves capacitats multimèdia amb velocitats de transferència de dades molt superiors en comparació a l'anterior generació, permetent videotrucades o una correcta connexió a Internet.

En aquest context va arribar la quarta generació amb la intenció de millorar l'estàndard UMTS, simplificar la xarxa d'accés de radi i millorar les velocitats de transferència de dades.

2.2.2. Introducció a la tecnologia LTE

LTE presenta una arquitectura simplificada en comparació amb les tecnologies anteriors. Està separada per dos blocs ben definits: CN (*Core Network* o Nucli de la Xarxa) i RAN (*Radio Access Network* o la Xarxa d'Accés de Radi). Comunament es coneixen aquests blocs amb el nom d'EPC (*Evolved Packet Core* o Nucli de Paquet Evolucionat) i E-UTRAN (*Evolved Universal Terrestrial Radio Access Network* o Xarxa Universal d'Accés Radi Terrestre Evolucionat) respectivament. El conjunt format per aquests dos elements es denomina com EPS (*Evolved Packet System* o Sistema de Commutació de Paquets Evolucionat).

Els objectius principals per a desenvolupar aquesta tecnologia varen ser:

- Necessitat d'assegurar la continuïtat de la competitivitat del sistema 3G per al futur.
- Demanda dels usuaris de majors velocitats de dades i qualitat de servei.
- Optimitzar la commutació de paquets.
- Reducció de costos.
- Baixar la complexitat.

- Evitar la fragmentació innecessària de les tecnologies.

Amb els anteriors objectius com filosofia el disseny d'aquesta tecnologia es va centrar en la reducció de nodes buscant una solució on la RAN estigués formada per un sol node i que el CN sigui el més independent possible.

Aquesta filosofia ha conduït a la implementació d'un node més complex que el NodeB d'UMTS, l'eNB (*evolved Node Base* o Node Base evolucionat) que té la funció de gestionar els recursos de radi i la connexió d'UEs a la xarxa.

El nucli de la xarxa també ha sofert d'aquesta filosofia i s'ha minimitzat el seu nombre de nodes. Aquest nucli està basat en el sistema UMTS i és conegut com a EPC que és responsable del pla de control i del pla d'usuari o de l'encaminament de les dades.

Al sistema EPS, tant els serveis a temps real com els serveis de dades són transportats pel protocol IP, és a dir, aquesta tecnologia està basat purament en IP. La direcció IP s'assigna quan l'UE es connecta a la xarxa i s'allibera quan es desconnecta.

LTE es basa en l'OFDMA (*Orthogonal Frequency Division Multiple Access* o Accés Múltiple per Divisió de Freqüències Ortogonals), en la combinació amb una modulació de fins a 64 QAM (*Quadrature Amplitude Modulation* o Modulació d'Amplitud en Quadratura) i grans amplades de banda de fins a 20 MHz.

A la figura 2.1 es pot observar els canvis produïts de generació en generació a l'arquitectura de les xarxes mòbils:

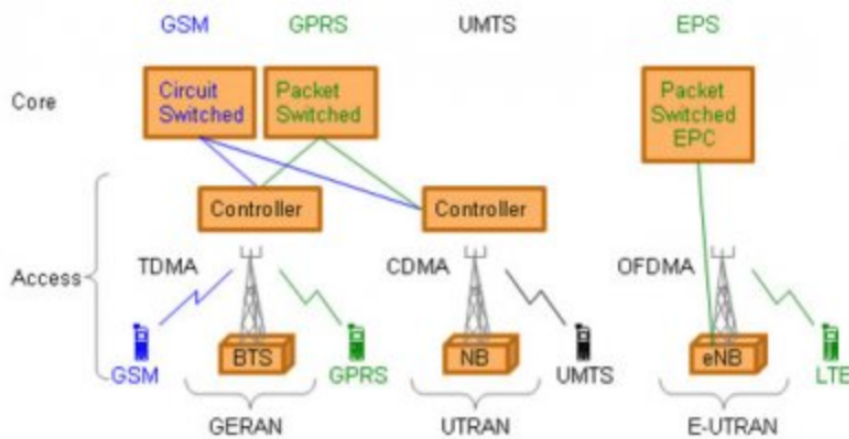


Figura 2.1.: Evolució de les xarxes mòbils [38].

Als següents apartats s'estudien, d'una forma més profunda, les funcionalitats dels elements que conformen l'arquitectura de la xarxa LTE.

2.2.3. Arquitectura de xarxa: eNB

El *evolved Node Base* (eNB) constitueix l'estació base de l'E-UTRAN integrant tota la funcionalitat de la xarxa d'accés. El BSC (*Base Station Controller* o Controlador de l'Estació Base) i el RNC (*Radio Network Controller* o Controlador de la Xarxa de Radi) heretats de generacions anteriors desapareixen de l'E-UTRAN i l'eNB guanya aquestes funcionalitats i passa a estar directament connectat amb la xarxa central fent servir la interfície S1.

Com que l'eNB no gaudeix d'una unitat central per combinar tota la informació dels UEs s'ha definit una nova interfície X2 que funciona com si fos una malla, és a dir, connecta diferents eNB entre si. El propòsit principal d'aquesta interfície és minimitzar la pèrdua de paquets a causa de la mobilitat de l'usuari. A mesura que l'usuari es mou, els paquets que encara no han sigut enviats o tractats per una estació base antiga poden ser enviats a una estació base nova.

L'eNB s'encarrega de la transmissió de paquets IP cap a i des dels equips de l'usuari fins al nucli de la xarxa EPC. Envia informació de l'usuari i missatges de senyalització. Les funcionalitats més característiques són:

- Modulació i desmodulació.
- Codificació i decodificació de canal.
- Control dels recursos de radi.
- Gestió de la mobilitat de radi.
- Detecció i correcció d'errors a la capa física.

A la següent figura es pot observar l'estructura que componen els eNodeB mitjançant la interfície X2 on comparteixen les dades de l'usuari i la seva connexió amb el nucli de xarxa (EPC) mitjançant la interfície S1:

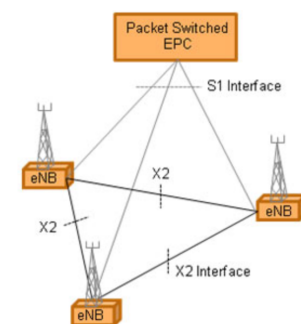


Figura 2.2.: Estructura de la malla d'eNBs [38].

2.2.4. Arquitectura de xarxa: EPC

El disseny de l'EPC es basa principalment a proporcionar un servei de connectivitat IP (evolució del GPRS). Està compost per una xarxa de servidors i encaminadors que s'encarreguen de la gestió del tràfic de la xarxa, treballa únicament en la commutació de paquets.

El nucli de la xarxa està compost pels diversos elements principals:

- MME (*Mobility Management Entity* o Entitat de Gestió de la Mobilitat).
- HSS (*Home Subscriber Server* o Servidor de Subscripcions).
- S-GW (*Serving Gateway* o Encaminador/Porta d'Enllaç Servidora).
- PDN Gateway (*Packet Data Network* o Xarxa de Dades de Paquets).
- PCRF (*Policy and Charging Rules Function* o Funció de la Política i les Normes de Cobrament).

A la figura 2.3 es pot observar una visió general dels diferents elements que componen l'arquitectura de l'EPC:

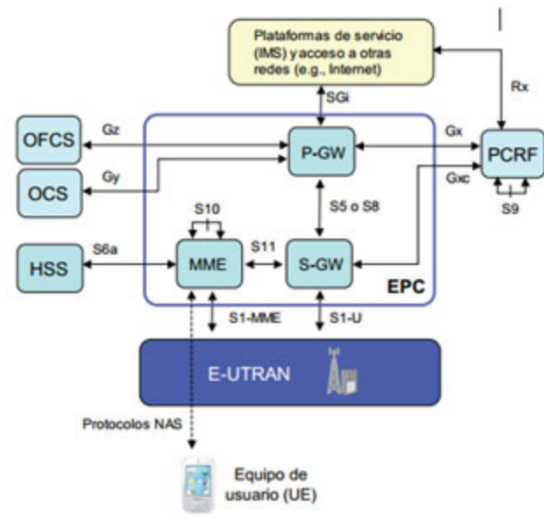


Figura 2.3.: Estructura de l'EPC [39].

A continuació s'expliquen els elements rellevants en la creació del prototip: MME, HSS, S-GW i P-GW.

MME

L'MME s'encarrega de totes les funcions del pla de control relacionades amb la gestió d'abonats i sessions. Les funcions que té són:

- Procediments de seguretat: autenticació de l'usuari, algorismes de xifrat i protecció de la integritat.
- Control de les sessions dels usuaris a la xarxa: procediments a la senyalització empleats per a establir el context de les dades en paquets i la negociació de la qualitat del servei.
- Gestió de la ubicació d'usuaris i procediments de mobilitat.

L'MME està vinculat a l'HSS mitjançant la interfície S6 que conté la base de dades on es troba tota la informació de les subscripcions de l'usuari.

HSS

L'HSS uneix el HLR (*Home Location Register* o Registre de Localització) i l'AuC (*Authentication Center* o Centre d'Autenticació) de generacions (3G i 2G) anteriors per a formar una base de dades que conté totes les subscripcions de l'usuari. Les funcions principals són:

- Identificació i direcció de l'usuari: correspon a l'IMSI (*International Mobile Subscriber Identity* o Identitat Internacional del Suscriptor Mòbil) i al MSISDN (*Mobile Subscriber ISDN Number* o Número de l'Estació Mòbil en la Xarxa Digital de Serveis Integrats) o número de telèfon mòbil.
- Informació del perfil de l'usuari: inclou els estats de subscripció al servei i de la seva qualitat (tassa màxima de bits permesa, classe de tràfic permesa, etc.).

També genera informació de seguretat per l'autenticació mútua entre l'usuari i la xarxa, la protecció de la integritat i el xifrat de la ruta de radi. Està connectat a l'MME per la interfície S6.

S-GW

El S-GW és un encaminador o porta d'enllaç, és a dir, és el punt de terminació de la interfície de dades de paquets fins E-UTRAN. És l'encarregat d'encaminar els paquets entre els eNBs dins d'una xarxa E-UTRAN quan els terminals es mouen a través de la xarxa. Té la funció de servir com una àncora de mobilitat local.

Està connectat a l'E-UTRAN mitjançant la interfície S1-U (la U indicia pla d'usuari: tràfic de dades d'usuari) i al P-GW per les interfícies S5 o S8.

P-GW

El P-GW té una funció pareguda al S-GW, també és un punt de terminació de la interfície de dades de paquets, però aquest cop es connecta a internet. Fa de punt d'ancoratge entre les sessions d'usuari fins xarxes de dades de paquets externes (internet). També permet aplicar les normes definides per l'operador per l'assignació i ús de recursos així com el filtratge de paquets.

Està connectat a internet mitjançant la interfície SGi i al S-GW per la interfície S5 o S8.

2.2.5. Protocols de xarxa

L'arquitectura de protocols de xarxa de LTE es pot separar en pla de control i pla d'usuari. Tal com es pot veure a la figura 2.4:

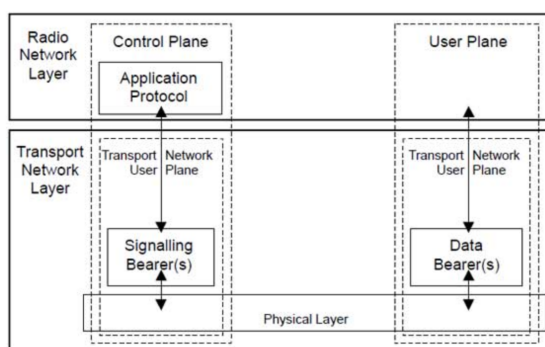


Figura 2.4.: Estructura de protocols LTE [19].

Al pla d'usuari la comunicació es realitza a través de paquets de dades processats per protocols IP, TCP (*Transmission Control Protocol* o Protocol de Control de Transmissió) i UDP (*User Datagram Protocol* o Protocol de Datagrames d'Usuari). I, al pla de control, el RRC (*Radio Resource Control* o Protocol de Recursos Radioelèctrics) crea els missatges de senyalització que s'intercanvien l'estació base i l'UE.

Tant al pla d'usuari com al pla de control la informació és processada pels protocols PDCP (*Packet Data Convergence Protocol* o Protocol de Control dels Paquets de Dades), RLC (*Radio Link Control* o Control de l'Enllaç Radi) i MAC (*Medium Access Control* o Control d'Accés al Medi) abans de passar a la capa física per la seva transmissió.

Pla d'usuari

El pla d'usuari s'encarrega dels protocols utilitzats per l'enviament de paquets IP corresponent a serveis als quals accedeixen els UEs a través de la xarxa. Perquè aquest l'intercanvi d'informació sigui possible s'utilitza un *stack* (o pila) de protocols on cada un té una funció específica. Les principals funcions d'aquests protocols es defineixen a continuació:

- PDCP: constitueix la capa superior i s'encarrega de rebre i d'enviar els paquets IP del tràfic d'usuari. Té la funció de la comprensió de capçaleres dels paquets IP i el xifrat de la informació per a garantir la seva seguretat i confidencialitat. També afegeix una capçalera amb un número de sèrie únic que permet efectuar una entrega organitzada dels paquets.
- RLC: permet enviar de forma segura els paquets PDCP entre l'eNB i l'UE. Té funcions de correcció d'errors mitjançant mecanismes ARQ (*Repeat ReQuest* o Petició Repetida), contenització, segmentació, detecció de duplicats i recuperació d'errors al protocol.
- MAC: s'encarrega de controlar l'accés al canal de radi. Suporta funcions de *scheduling* (o programació) dinàmiques entre UEs controlant prioritats i multiplexant paquets RLC de diferents serveis de portadors de radi als canals de transport de la capa física.

Entre l'eNB i l'EPC tot el tràfic d'usuari s'encapsula mitjançant el protocol GTP (GPRS *Tunneling Protocol* o Protocol de Tunelat GPRS) basat en el protocol IP para el nivell de xarxa i el protocol UDP pel nivell de transport.

L'arquitectura dels protocols pel pla d'usuari es pot observar en la figura 2.5.

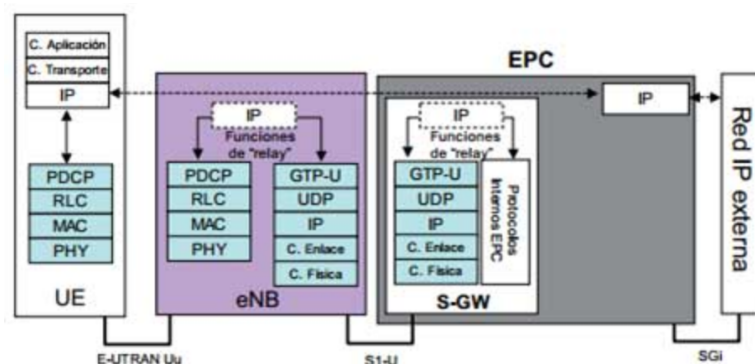


Figura 2.5.: Estructura de protocols del pla d'usuari [40].

Pla de control

El pla de control es refereix als protocols necessaris per a sustentar les funcions i procediments en les diferents interfícies. Aquest pla també es suporta sobre el mateix *stack* que el pla d'usuari (protocols PDCP, RLC, MAC i la mateixa capa física) però afegeix dos protocols nous en l'àmbit de xarxa:

- RRC (*Radio Resource Control* o Control de Recursos de Radi): permet establir una connexió de control entre l'eNB i l'UE per a dur a terme funcions de gestió de

la interfície de radi. Gestiona serveis de portadors de radi, funcions de mobilitat i la difusió de paràmetres de sistema.

- NAS (*Non Access Stratum* o Estrat de No Accés): els protocols NAS intercanvia informació entre l'entitat MME de l'EPC i l'UE. Les principals funcions són d'autenticació, autorització i gestió de mobilitat dels terminals que no tenen una connexió RRC establerta.

L'arquitectura dels protocols pel pla de control es pot observar en la figura 2.6.

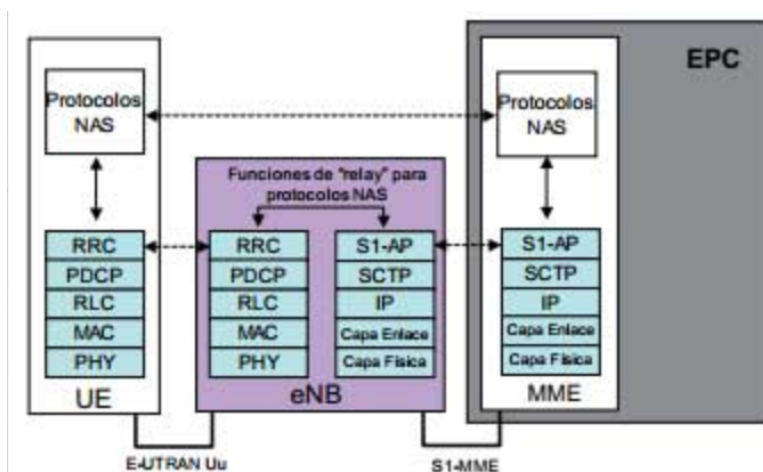


Figura 2.6.: Estructura de protocols del pla de control [40].

2.3. SDR (*Software Defined Radio*)

2.3.1. Descripció general

Un SDR (*Software Defined Radio* o Ràdio Definida per *Software*) és un sistema de radiocomunicacions on alguns dels components típicament implementats en *hardware* (mescladors, filtres, moduladors/desmoduladors, detectors, etc) són implementats en *software* a un ordinador o un dispositiu embegut.

Un SDR bàsic pot ser un ordinador personal amb una targeta de so i un ADC (*Analog-to-Digital Converter* o Convertidor Analògic-Digital) precedit per un adaptador de radiofreqüència (RF). Per tant, només utilitzant un ADC, un DAC (*Digital-to-Analog Converter* o Convertidor Digital-Analògic) i les antenes poden realitzar les conversions dels senyals analògics a digital i viceversa. Això fa que els SDR siguin molt flexibles per a dur a terme diferents funcions, a més de facilitar la solució de problemes.

Els SDR tenen una gran utilitat per usos militars i serveis de telefonia mòbil que han de servir una àmplia varietat de protocols de radi canviant a temps real. S'espera que es converteixin en la tecnologia dominant en les comunicacions per radi.

2.3.2. USRP (*Universal Software Radio Peripheral*)

Un USRP (*Universal Software Radio Peripheral* o *Software Universal Perifèric de Ràdio*) és un tipus de SDR creat per la companyia *Ettus Research* pertanyent a l'empresa *National Instruments*. La família USRP va ser dissenyada per l'accessibilitat, i molts dels productes són *hardware* de codi obert.

La majoria dels USRP es connecten a un ordinador central mitjançant un enllaç d'alta velocitat com, per exemple USB 3.0, que el *software* del *host* (o amfitrió) utilitza el *hardware* de l'USRPs per a rebre o transmetre dades.

Els USRP proporcionen una solució en el disseny per a crear ràpidament prototips de sistemes de comunicacions sense fils. Presenten aplicacions amb senyals d'entrada múltiple, sortida múltiple (MIMO) i bancs de proves per LTE i WiFi. Estan dissenyats per a poder servir a aplicacions que operen entre DC *Direct Current* (o Corrent Continu) (0 Hz) fins a 6 GHz

Tota la família de dispositius USRP es controlen mitjançant el controlador UHD (*Ettus Research Universal Hardware Driver* o Control Hardware Universal de *Ettus Research*).

2.4. Programari de codi obert

En aquest apartat es presenten el programari de codi obert que s'ha usat per a la creació del dispositiu i la seva relació.

2.4.1. *Open-Air-Interface* (OAI)

La creació d'OAI es va dur a terme en 2014 quan EURECOM va crear l'OSA (*OpenAir Interface Software Alliance*). Una organització sense fins lucratiu que EURECOM lidera.

Open-Air-Interface és una plataforma de codi obert que implementa les especificacions LTE i 5G del 3GPP. Proporciona *software* per la xarxa troncal (EPC) i per la xarxa d'accés (E-UTRAN) desenvolupades en el llenguatge de programació C.

La plataforma està dividida en dues parts:

- *OpenairCN*: implementa els diferents elements de l'EPC: MME, HSS i SPGW que agrupa els mòduls S-GW i P-GW.
- *Openair5G*: implementa la xarxa d'accés (E-UTRAN), definint els eNB. A més, d'implementar UEs.

Aquests projectes presenten una compatibilitat amb dispositius comercials, és a dir, es pot usar una estació base implementada amb OAI i que l'EPC i l'UE siguin comercials. També es pot usar un UE d'OAI amb un eNB comercial.

Finalment, OAI presenta molta flexibilitat gràcies a la comunitat que s'ha format al seu voltant i dels múltiples tutorials que disposa al públic donant moltes possibilitats a l'hora de realitzar muntatges.

2.4.2. Mosaic5G

Mosaic5G es va crear l'any 2016 i és una iniciativa sense ànim de lucre que fomenta el desenvolupament de *software* de codi obert amb fi de realitzar un servei de xarxa 5G.

Mosaic5G proporciona quatre plataformes principals:

- *LL-MEC*: implementa un controlador de xarxa mòbil.
- *FlexRAN*: implementa un controlador de xarxa per una plataforma RAN definida.
- *JOX*: implementa un orquestrador per a gestionar les xarxes 5G basat en *JuJu*.
- *Kube5G*: implementa una plataforma de servei àgil 4G/5G a la núvol.

El projecte pel qual s'utilitza al prototip és *FlexRAN* que és l'eina que permet fer el monitoratge i la divisió de l'espectre (*slicing*).

Totes aquestes plataformes estan creades sobre els projectes de RAN (*Openair5G*) i CN (*OpenairCN*) que propociona OAI per així poder permetre la supervisió, el control i la programabilitat de cada mòdul de RAN i CN de la xarxa. Per tant, encara que es treballi sobre els arxius de configuració obtinguts de Mosaic5G realment són els que OAI ha implementat, és a dir, per a configurar l'eNB i l'EPC s'usen els projectes d'OAI.

2.4.3. Open Cells

Open Cells és una companyia creada i dirigida per Laurent Thomas. El seu objectiu és transformar el codi obert d'OAI de la *Open-Air Software Alliance* a una implementació de referència pública dels estàndards sense fils.

Open Cells també ha participat en la creació del codi obert d'OAI. Les contribucions més importants que presenta són:

- Turbo codificador i descodificador: ha augmentat l'eficiència de la CPU (*Central Processing Unit* o Unitat de Processament Central) en 10 vegades en comparació de l'anterior implementació realitzada per OAI. Això permet que el sistema funcioni en temps real per a desenes de Mb/s.
- Millores del *software* d'OAI en temps real.
- Eines de compilació i generació de codi: compilació basada en procediments de *cmakes*.
- Generació de *targets* (o objectius) en directoris fora del codi font.

També es dediquen a la distribució de targetes SIM per a ser usades en entorns de xarxa creats a partir dels codis d'OAI. Al projecte s'utilitza una targeta SIM d'*Open Cells* per tal de connectar l'UE amb la xarxa. Com es veurà a l'apartat 4 també disposen al públic d'una base de dades on es troba tota la informació de les targetes.

2.5. MySQL i phpMyAdmin

MySQL és un base de dades relacional de codi obert del tipus SQL (*Structured Query Language* o Lenguatge de Consultes Estructurat) amb un model servidor-client. Va ser creada a l'any en 1994 per una companyia sueca anomenada *MySQL AB*. Avui dia és la companyia *Oracle* l'encarregada del seu desenvolupament, mantingut i distribució.

phpMyAdmin és una eina escrita en PHP (*HyperText Preprocessor* o Preprocessador d'Hipertext) amb la intenció de manejar l'administració de MySQL a través de pàgines web. És un projecte que està vigent des de 1998 i pot crear i eliminar bases de dades, crear, eliminar i alterar taules, esborrar, editar i afegir camps, executar qualsevol sentència SQL, administrar claus en camps, administrar privilegis i exportar dades en diversos formats.

2.6. Eines de verificació i monitoratge de la xarxa

En aquesta secció s'explica de forma breu les diferents eines que s'han utilitzat al projecte per tal de comprovar el bon funcionament de la xarxa i monitorar-la.

2.6.1. *iPerf3*

iPerf3 és una eina que s'utilitza per a la realització de proves de xarxes informàtiques. El funcionament és la creació de dades TCP o UDP per a mesurar el rendiment de la xarxa. Està basat en el seu avantpassat *iPerf*.

Està pensat perquè funcioni com client i com servidor i, així, mesurar el rendiment dels dos extrems de la xarxa tant unidireccionalment com bidireccionalment. És un *software* de codi obert que pot ser executat en *Linux*, *Unix* i *Windows*.

2.6.2. *Wireshark*

Wireshark és un analitzador de protocols empleat per a la realització d'anàlisis i solucionar problemes de xarxes de comunicació. També s'usa per a l'anàlisi de dades i protocol i com una eina didàctica. Té una interfície gràfica i disposa de moltes opcions de filtratge i organització de la informació, d'aquesta forma permet veure tot el tràfic de dades d'una xarxa.

Wireshark és un *software* lliure que es pot executar sobre *Linux*, *MacOS* *Unix* i *Windows*.

2.6.3. *FlexRAN*

FlexRAN és una plataforma de monitoratge que es compon de dos elements principals: pla de servei i control i pla d'aplicació. La separació en dos plans és proporcionat per l'entorn de temps d'execució de la RAN que actua com una capa d'abstracció que

separa el mòdul RAN, el RTC (*Real Time Controller* o Controlador en Temps Real) i les aplicacions de control.

El pla de servei i control segueix un disseny jeràrquic i està compost per RTC que està connectat a un número de temps real d'execució de la RAN. El protocol *FlexRAN* facilita la comunicació en el controlador en temps real i l'agent RAN incorporant en l'entorn de temps d'execució.

Les característiques principals que té *FlexRAN* són:

- Control de la RAN i separació del pla de dades: desuneix el pla de control i de dades proporcionant beneficis com la reducció de la complexitat a l'hora de desenvolupar noves solucions de control.
- Control centralitzat i en temps real: consolida el pla de control en únic controlador centralitzat que permet una coordinació més fàcil les estacions base.
- Funcions d'abstracció i control virtualitzat: permet el control flexible i programable de la infraestructura de la RAN.
- Delegació de control i reconfiguració de polítiques: permet la delegació de les funcions de control, com les programadores i els gestors de mobilitat, del controlador mestre a les estacions base en temps d'execució i la reconfiguració del seu comportament i paràmetres sobre la marxa de manera senzilla.

I les aplicacions principals són:

- Optimització de la RAN: anàlisi de dades, divisió de la RAN amb consciència de la qualitat del servei, gestió de mobilitat, control d'interferències, etc.
- Optimització del contingut: optimització a l'hora de consumir contingut multimèdia, creació de perfil i serveis, etc.
- Optimització del dispositiu/proveïdor: gestió i distribució de l'espectre (*slicing*) i control d'accés a la *slice*, posicionament, etc.

Al projecte s'usa *FlexRAN* pel monitoratge a temps real de la RAN i la creació de *slices* i la distribució dels recursos de la xarxa.

3. Especificacions i restriccions del muntatge

En aquest apartat s'explica les especificacions i les restriccions que té el disseny del prototip de xarxa a desplegar. L'arquitectura del projecte es veu a la figura 3.1:

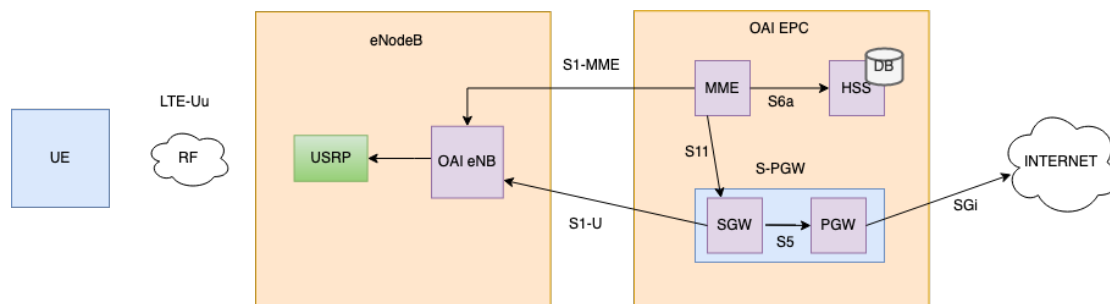


Figura 3.1.: Arquitectura de la xarxa a desplegar.

3.1. Especificacions

- S'utilitza la plataforma de codi lliure Mosaic5G que disposa de tot el material necessari per a dur a terme el projecte. D'aquesta forma tampoc s'ha de pagar cap llicència.
- El prototip està dividit en dues parts diferenciades per tal de complir amb les especificacions de l'arquitectura de xarxa LTE. Una part fa les funcions de la xarxa d'accés de radi (E-UTRAN) i l'altra de nucli de la xarxa (EPC).
- S'usa un dispositiu SDR mòdel USRP-B205mini i una estació de treball de sobretaula per a realitzar les funcions encarregades de la xarxa d'accés de radi.
- L'estació de treball és un ordinador de sobretaula mini PC GIGABYTE i7.
- L'eNB es troba al sistema operatiu mare de l'ordinador empleat i l'EPC està a una maquina virtual.
- Tots els mòduls de *software* i les interfícies de xarxa empleades per a la comunicació dels mòduls han estat implementades per OAI i obtingudes mitjançant els repositoris de Mosaic5G. Al projecte només es configuren els fitxers dels mòduls de *software* per tal de crear una xarxa LTE.

- El sistema operatiu de l'eNB i de l'EPC és Ubuntu 16.04.
- El *kernel* de l'eNB ha de ser *low-latency* i el *kernel* de l'EPC ha de tenir el mòdul GTP (*GPRS Tunneling Protocol*).
- L'eNB i l'EPC es comuniquen mitjançant la interfície de xarxa que crea la màquina virtual. És l'encarregada de realitzar l'enviament de paquets de les interfícies de xarxa LTE S1-MME i S1-U.
- L'estació de treball està connectada a internet mitjançant un cable *ethernet*.
- El registre d'usuaris es fa per mitjà de la base de dades MySQL.
- Per a connectar l'UE a la xarxa s'empra una targeta SIM d'*Open Cells* inserida dins d'un *dongle*.
- El monitoratge de la xarxa i la divisió d'espectre es realitzen via *FlexRAN*.

3.2. Restriccions

- Per a dur a terme el projecte OAI demana que s'utilitzin processadors i5 o i7 de tercera generació o posterior, Intel Xenon de segona generació o posterior o Intel Atom Rangeley.
- OAI recomana la instal·lació de *Kernels* de *low-latency* (o de baixa latència) per a dur a terme les funcions de l'E-UTRAN.
- No es poden realitzar trucades perquè OAI no ha implementat aquesta funcionalitat.
- No es pot ni executar ni configurar l'element de l'EPC PCRF perquè OAI no ho ha implementat.
- No es poden realitzar proves de *handover* o *paging* perquè no es troben en la integració d'OAI.
- La configuració en mode FDD de la xarxa d'accés només és estable amb amplades de banda de 5 o 10 MHz.

4. Descripció del muntatge

4.1. Descripció general

En aquesta secció es presenta el muntatge del prototip de la xarxa 4G-LTE implementada amb OAI i Mosaic5G. A la següent figura es pot observar l'arquitectura del projecte:

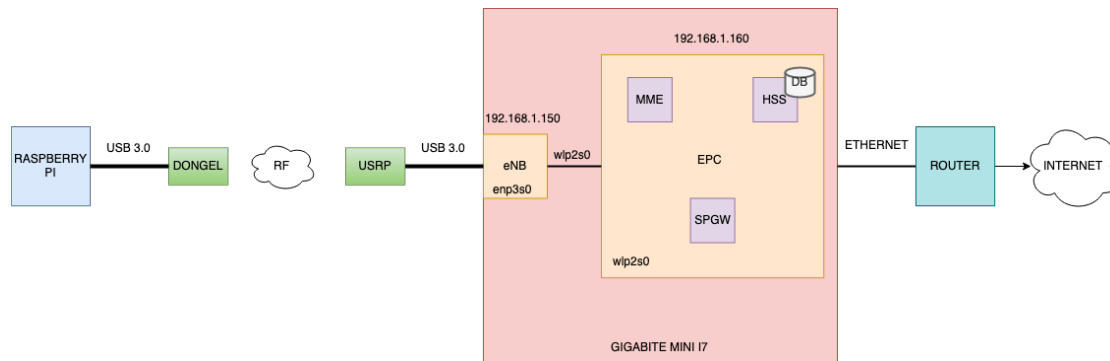


Figura 4.1.: Arquitectura de la xarxa.

El *dongle* (que té una SIM inserida) és l'UE del prototip que fa de *hotspot* a una *Raspberry*. Les dades de la xarxa li arriben per radiofreqüència (RF) enviades per l'USRP. Aquest últim està connectat a l'eNB mitjançant un USB 3.0. L'eNB està instal·lat al sistema operatiu natiu de l'ordinador Gigabyte Mini i7 i l'EPC es troba a una màquina virtual. Tant la màquina virtual com el sistema operatiu natiu de l'estació són Ubuntu 16.04. Finalment, l'ordinador està connectat al *router* (o encaminador) mitjançant un cable *ethernet* que enllaça tot el sistema a internet.

L'eNB fa el paper de xarxa d'accés de radi (E-UTRAN, eNodeB) de LTE. Està format pel propi eNB i per l'USRP. L'eNB és un mòdul de software que emula el funcionament d'una estació base i l'USRP és un dispositiu SDR que s'encarrega de la transmissió i la recepció de dades entre la xarxa i l'UE.

L'EPC es correspon amb el nucli de la xarxa LTE que implementa els mòduls MME, HSS i SPGW corresponents a una xarxa de quarta generació. L'SPGW fa referència a les sigles (*Serving and Packet Data Network Gateway* o Encaminador/Porta d'Enllaç Servidora i PDN o Internet) sent un bloc de software que emula les funcionalitats de S-GW i P-GW del nucli de xarxa LTE. Cal destacar que la base de dades (BD) es troba a l'HSS.

Tant l'eNB com l'EPC tenen una IP fixa que en aquest cas són: 192.168.1.150/24 i 192.168.1.160/24, respectivament i es comuniquen fent servir la interfície de xarxa *wlp2s0*.

A una xarxa LTE es poden usar diferents bandes de freqüència per la transmissió de dades. Al projecte s'utilitza la banda 7 pel fitxer de configuració que proporciona Mosaic5G (ja té els valors per defecte). Consta de les següents característiques principals:

E-UTRA Band	Duplex-Mode	Freqüència (MHz)	Nom Comú
7	FDD	2600	IMT-E

Taula 4.1.: Especificacions de la banda 7.

4.2. Configuració inicial de l'escenari

Abans de poder començar amb la configuració dels diferents components de la xarxa s'ha de fer una posada a punt a l'estació de treball Gigabyte Mini i7. Sense aquest pas previ no es pot dur a terme el projecte.

4.2.1. Configuració inicial de la màquina eNB

En aquesta secció es veu la instal·lació i configuració inicial de l'eNB que executa el software d'*Open-Air-Interface* (OAI). Tot el software del qual es parla fa referència al projecte OAI-RAN extret de Mosaic5G.

El primer pas consisteix en la instal·lació d'Ubuntu a la versió de 16.04 de 64 bits, sistema operatiu que OAI demana.

En aquesta màquina es du a terme el processament dels senyals de radiofreqüència tant de transmissió com de recepció. Per aquest motiu es requereix una baixa latència i que tots els recursos físics de l'estació estiguin disponibles per a poder dur a terme els càlculs a temps real. Per tant, es realitza la instal·lació d'un *kernel* de *lowlatency* (o de baixa latència) al sistema operatiu natiu. Per Ubuntu 16.04, OAI recomana la versió 4.4.0-21 de *kernel-lowlatency*.

El següent pas consisteix a deshabilitar les funcions que permeten al sistema operatiu gestionar el consum d'energia i controlar la potència del processador. D'aquesta forma s'aconsegueix que la CPU (*Central Processing Unit* o Unitat de Processament Central) sempre estigui funcionant al màxim de les seves especificacions. A més, per assegurar que l'estació de treball usi tots els recursos en el processament dels senyals de radiofreqüència es desactiva l'*Hyper-Threading*. L'*Hyper-Threading* és una implementació de *hardware* que permet que cada nucli del processador pugui dur a terme dos processos a l'hora. Per a realitzar aquests canvis s'ha de modificar l'arxiu *grub* del sistema, que es troba al *path* (o camí) `/etc/default/grub`.

També es pot canviar el *hostname* (nom de l'amfitrió) de l'ordinador per *enb* i s'han de desactivar les actualitzacions automàtiques per evitar problemes de dependències a l'hora de la configuració de la xarxa.

L'últim pas consisteix a definir una IP fixa, que en el cas de l'eNB, és 192.168.1.150 a l'interfície de xarxa *enp3s0*.

El procediment detallat per a fer tots aquests passos es pot consultar a l'ANNEX A.

4.2.2. Configuració inicial de la màquina EPC

En aquesta secció es veu la instal·lació i configuració inicial de l'EPC que executa el software d'*Open-Air-Interface* (OAI). Tot el software del qual es parla fa referència al projecte OAI-CN extret de Mosaic5G.

Com que al projecte s'usa la mateixa màquina per executar eNB i EPC es pot instal·lar una màquina virtual (MV) de *VMware* per l'EPC i així tenir un muntatge més senzill. El motiu de la instal·lació d'una màquina virtual (i no un contenidor) és la necessitat d'executar dos *kernels* diferents i, una MV és l'única forma d'instal·lar dos sistemes operatius diferents sense compartir el *kernel*.

La MV constarà de les següents especificacions principals:

Nº de Processadors	Memòria	Espai de disc	Tipus de Xarxa
2	4GB	80GB	Bridge

Taula 4.2.: Especificacions de la Màquina Virtual.

Una vegada creada la MV s'han de fer uns canvis al seu sistema operatiu perquè pugui dur a terme el seu paper d'EPC. S'ha d'instal·lar un *kernel* que disposi del mòdul GTP (GPRS *Tunneling Protocol* o Protocol de tunelització del GPRS) per a poder encapsular tot el tràfic pla de l'usuari. Ubuntu, a partir del *kernel* 4.7.0, ja disposa d'aquest mòdul, per tant, s'instal·la qualsevol *kernel* superior a aquesta versió.

El següent pas consisteix a canviar el *hostname* de la màquina. Aquest cas el nom és *epc*. També s'ha de modificar el fitxer */etc/hosts* per a poder donar-li un *realm* (o àmbit) a la màquina. Aquest pas és important perquè a l'EPC funcionen a l'hora tres mòduls de software. Per a la correcta comunicació entre ells és necessari crear una identitat que ve definida pel *hostname* i el *realm*. A aquesta identitat se li anomena FQDN (*Full Qualified Domain Name* o Nom de Domini Completament Qualificat) i té la forma: *hostname.realm*. A aquest projecte el *realm* escollit és *openair4G.eur*. Per tant, el FQDN del projecte és: *epc.openair4G.eur*. Aquests mòduls, en l'àmbit d'implementació d'OAI utilitzen *FreeDiameter* per comunicar-se. Cal destacar que tant el *hostname* com el *realm* es poden anomenar de qualsevol manera.

Finalment, es defineix una IP fixa, que al cas de l'EPC, és 192.168.1.160 a la interfície de xarxa *wlp2s0*. També es desactiven les actualitzacions automàtiques per a evitar problemes de dependències.

El procediment detallat per a fer tots aquests passos es pot consultar a l'ANNEX B.

4.3. Instal·lació de Mosaic5G

Una vegada que es finalitza la configuració inicial es comença el procés d'instal·lació dels mòduls de *software* necessaris a cada màquina per a dur a terme el projecte.

Per a obtenir els mòduls, Mosaic5G, té dues opcions:

- Descarregar el *git* del projecte de Mosaic5G.
- Obtenir els codis mitjançant *snap*.

Al prototip s'utilitza la segona opció perquè per la primera s'ha de tenir un compte a OAI i no es disposa de cap.

La instal·lació comença comprovant si hi ha actualitzacions al sistema amb la comanda:

```
sudo apt update
```

Quan s'han instal·lat les actualitzacions (si hi havia), s'instal·la *snap* per Ubuntu:

```
sudo apt install snapd
```

Un cop que s'ha instal·lat correctament *snap* s'han de descarregar uns paquets previs tant a l'eNB com a l'EPC abans d'obtenir els mòduls de software necessaris per a la realització del prototip. Aquest paquet són l'editor de text *emacs*, les eines de monitoratge *iPerf3* i *Wireshark* i l'*openssh-server* per a tenir la possibilitat d'accedir remotament a l'ordinador i facilitar el treball. La comanda és:

```
sudo apt install emacs iperf3 wireshark openssh-server
```

El següent pas és obtenir els mòduls de software. A l'eNB s'instal·la *FlexRAN* i OAI-RAN que fa la funció d'eNB. Les comandes són:

```
sudo snap install flexran --channel=edge  
sudo snap install oai-ran --channel=edge
```

Amb l'opció de *-channel=edge* s'obté l'última versió dels mòduls de software.

Finalment a l'EPC s'instal·la OAI-CN que fa les funcions de nucli de xarxa:

```
sudo snap install oai-cn --channel=edge
```


4.4. Descripció del muntatge

En aquest apartat s'indiquen les tasques necessàries per a dur a terme el prototip, començant amb el nucli de xarxa (EPC), seguit de l'estació base (eNB), la base de dades dels usuaris i, finalment el *dongle*, que dona connexió a internet a la *Raspberry*.

Un cop que s'ha arribat a aquest punt es configura el *FlexRAN* i, utilitzant-ho, es divideix l'espectre en diferents *slices* per a donar més o menys recursos a l'UE.

4.4.1. Configuració de l'EPC

A continuació es veu la configuració completa del nucli de xarxa, es comença amb l'HSS, es segueix amb l'MME i es finalitza amb l'SPGW.

Configuració de l'HSS

El primer pas en la configuració de l'HSS és instal·lar la base de dades (BD) que usa per a gestionar els usuaris. La BD que s'utilitza és MySQL i la comanda empleada per a obtenir-la és:

```
sudo apt install mysql-server mysql-client
```

L'usuari per defecte de MySQL és *root* (o *arrel*) i la contrasenya que s'usa en el projecte és *linux*.

Una vegada que es té la base de dades s'ha d'aconseguir els fitxers de configuració del mòdul. La comanda que s'empra és:

```
sudo oai-cn.hss-conf-get
```

Per la correcta configuració s'ha d'editar dos fitxers: */var/snap/oai-cn/current/hss.conf* i */var/snap/oai-cn/current/hss_fd.conf*. En el primer es dona conèixer les dades de la BD i al segon la identitat i el *realm* del projecte que ha de coincidir amb */etc/hosts*.

Al fitxer */var/snap/oai-cn/current/hss.conf*, els paràmetres més importants són:

- *MYSQL_server*: especifica la direcció IP on es troba la BD que empra HSS. Com és a la mateixa màquina, el valor és: 127.0.0.1, és a dir, *localhost*.
- *MYSQL_user*: especifica el nom de l'usuari per identificar-se en la BD. Com ja s'ha vist abans, aquest nom ja s'ha configurat amb la instal·lació de MySQL. Es va establir l'usuari *root*.
- *MYSQL_pass*: especifica la contrasenya de l'usuari per identificar-se en la BD. Com ja s'ha vist abans, aquest paràmetre ja s'ha configurat amb la instal·lació de MySQL. Es va establir la contrasenya *linux*.

- *MYSQL_db*: especifica el nom de la BD de OAI que utilitza HSS. Aquest valor es deixa per defecte *oai_db*. Si és diferent es posa un altre nom.
- *OPERATOR_key*: especifica el valor de la clau d'operació del HSS. En aquest cas, es deixa en blanc, perquè la targeta SIM d'*Open Cells* utilitzarà la seva pròpia clau.
- *FD_conf*: especifica la ruta del fitxer on es troba la configuració pel protocol *FreeDiameter*. Es configura amb el *path*: */var/snap/oai-cn/current/hss_fd.conf*.

Al fitxer */var/snap/oai-cn/current/hss_fd.conf*, es troben els següents paràmetres principals:

- *Identity*: especifica el FQDN que és *epc.openair4G.eur*.
- *Realm*: especifica el domini del FQDN que és *openair4G.eur*.
- Els altres paràmetres es deixen per defecte.

Un cop que els dos fitxers estan configurats es generen nous certificats amb el nom de la màquina amb la comanda:

```
sudo oai-cn.hss-init
```

A continuació, es comprova el funcionament del mòdul amb la comanda:

```
sudo oai-cn.hss
```

El resultat que s'ha d'obtenir pel terminal ha de ser el següent:

```
11:23:24 NOTI   Loaded extensions: 'dict_s6a.fdx'[(no config file)], loaded
11:23:24  DBG    Core state: 1 -> 2
11:23:24 NOTI   Local server address(es): 192.168.1.160{---L-}
11:23:24  DBG    Core state: 2 -> 3
Initializing s6a layer: DONE
```

Figura 4.2.: Resultat de l'execució de l'HSS.

En aquest punt l'HSS està esperant la connexió de l'MME.

Configuració de l'MME

El primer pas és obtenir els fitxers de configuració de l'MME:

```
oai-cn.mme-conf-get
```

Una vegada que ja es té els fitxers de configuració s'han d'editar dos arxius: `/var/snap/oai-cn/current/mme.conf` per la configuració general de l'MME i `/var/snap/oai-cn/current/mme_fd.conf` per la configuració del protocol *FreeDiameter*.

Al fitxer `/var/snap/oai-cn/current/mme.conf` es troben els següents paràmetres importants:

- *REALM*: especifica el domini del sistema. Es configura com *openoai4G.eur*.
- *SGA_CONF*: especifica la ruta on es troba el fitxer de configuració del protocol *FreeDiameter* que utilitza el MME. Es configura com `/var/snap/oai-cn/current/mme_fd.conf`.
- *HSS_HOSTNAME*: especifica el nom empleat al FQDN de l'HSS, en aquest cas, *epc*.
- *GUMMEI_LIST*: especifica la llista GUMMEI (*Globally Unique MME Identifier* o Identificador Global i Únic de l'MME). Està compost pel MCC (*Mobile Country Code* o Codi de País per a Mòbils), el MNC (*Mobile Network Code* o Codi de la Xarxa mòbil), el GUMMEI ID de l'MME i el codi de l'MME. Ha de coincidir amb els valors de l'eNB. La configuració és la següent:
 - *MCC*: 208.
 - *MNC*: 92.
 - *MME_GID*: 4.
 - *MME_CODE*: 1.
- *TAI_LIST*: especifica la llista de TAI (*Tracking Area Identity* o Identitat de l'Àrea de Seguiment). Està compost pel MCC, el MNC, i TAC (*Tracking Area Code* o Codi d'Àrea de Seguiment). Ha de coincidir amb els valors de l'eNB. La configuració és la següent:
 - *MCC*: 208.
 - *MNC*: 92.
 - *TAC*: 1.
- *MME_INTERFACE_NAME_FOR_S1_MME*: especifica el nom de la interfície de xarxa que utilitza l'MME per a simular la interfície S1-MME de LTE. Es configura amb *ens33*.
- *MME_IPV4_FOR_S1_MME*: especifica la direcció IP empleada per l'MME per a la interfície de xarxa S1-MME de LTE. Es configura amb 192.168.1.160/24.
- *MME_INTERFACE_NAME_FOR_S11_MME*: especifica el nom de la interfície de xarxa que utilitza l'MME per a simular la interfície S11-MME de LTE que connecta l'MME amb el SPGW. Es configura amb *lo*, ja que, els dos mòduls s'executen a la mateixa màquina i, per tant, és la interfície local.

- *MME_IPV4_FOR_S11_MME*: especifica la direcció IP empleada per l'MME per a la interfície de xarxa S11-MME de LTE. És la IP que utilitza l'MME per a comunicar-se amb el SPGW. Es configura amb 127.0.11.1/8 que és el valor per defecte.
- *SGW_IPV4_ADDRESS_FOR_S11*: especifica la direcció IP empleada per l'SPGW per a la interfície de xarxa S11-MME de LTE. És la IP que utilitza el SPGW per a comunicar-se amb l'MME. Es configura amb 127.0.11.2/8 que és el valor per defecte.
- Els altres paràmetres es deixen per defecte.

Al fitxer `/var/snap/oai-cn/current/mme_fd.conf`, es troben els següents paràmetres principals:

- *Identity*: especifica el FQDN que és *epc.openair4G.eur*.
- *Realm*: especifica el domini del FQDN que és *openair4G.eur*.
- Els altres paràmetres es deixen per defecte.

Un cop que els dos fitxers estan configurats es generen nous certificats amb el nom de la màquina amb la comanda:

```
sudo oai-cn.mme-init
```

A continuació, es comprova el funcionament del mòdul amb la comanda:

```
sudo oai-cn.mme
```

El resultat que s'ha d'obtenir pel terminal de l'MME és una taula que indica el nombre d'eNBs i d'UEs connectats. Tal com es veu a la següent figura:

```
000145 00000:483335 7FB0827FC700 ERROR SGA parts/mme/src/src/s6a/s6a_task.c:0080 Connected to 'epc.openair4G.eur' (TCP:sock#37), remote capabilities:
000146 00000:483341 7FB0827FC700 ERROR SGA parts/mme/src/src/s6a/s6a_task.c:0080 Capabilities-Exchange-Answer(237)[---], Length=200, Hop-By-Hop-Id=0x75f76136, End-to-End=0xc3382b49, {
ult-Code(268)[-M]=DIAMETER_SUCCESS(2001 (0x7d11)), { Origin-Host(264)[-M]=epc.openair4G.eur }, { Origin-Realm(296)[-M]=openair4G.eur }, { Origin-State-Id(278)[-M]=1607973804 (0x5fd7bbac)
[ Host-IP-Address(257)[-M]=192.168.1.160 }, { Vendor-Id(266)[-M]= (0x0) }, { Product-Name(269)[-M]=freeDiameter }, { Firmware-Revision(267)[-M]=10201 (0x27d9) }, { Vendor-Specific-Application
d(260)[-M]= { Auth-Application-Id(238)[-M]=16777231 (0x1000023) }, { Vendor-Id(266)[-M]=10415 (0x28af) }, { Supported-Vendor-Id(265)[-M]=10415 (0x28af) }
000147 00000:483349 7FB0827FC700 ERROR SGA parts/mme/src/src/s6a/s6a_task.c:0080 No TLS protection negotiated with peer 'epc.openair4G.eur'.
000148 00000:483304 7FB0827FC700 ERROR SGA parts/mme/src/src/s6a/s6a_task.c:0080 'STATE_WAITCEA' -> 'STATE_OPEN' 'epc.openair4G.eur'
000149 00001:435399 7FB0827FC700 DEBUG SGA parts/mme/src/src/s6a/s6a_peer.c:0115 Diameter identity of MME: epc.openair4G.eur with length: 17
000150 00001:436010 7FB0827FC700 DEBUG SGA parts/mme/src/src/s6a/s6a_peer.c:0150 S6a peer connection attempt 1 / 8
000151 00001:436019 7FB0827FC700 DEBUG SGA parts/mme/src/src/s6a/s6a_peer.c:0163 Peer epc.openair4G.eur is now connected...
000152 00010:363636 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0033 ===== STATISTICS =====
000153 00010:363643 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0034
000154 00010:363645 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0036 Connected eNBs | 0 | 0 | 0 |
000155 00010:363648 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0038 Attached UEs | 0 | 0 | 0 |
000156 00010:363652 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0040 Connected UEs | 0 | 0 | 0 |
000157 00010:363655 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0042 Default Bearers | 0 | 0 | 0 |
000158 00010:363657 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0044 S-U Bearers | 0 | 0 | 0 |
000159 00010:363659 7FB0A08F8700 DEBUG MME-AP src/mme_app/mme_app_statistics.c:0045 ===== STATISTICS =====
```

Figura 4.3.: Resultat de l'execució de l'MME.

I, al terminal de l'HSS, s'ha de veure com es passa de *STATE_CLOSED* a *STATE_OPEN*:

```

11:25:39 NOTI Connected to 'epc.openair4G.eur' (TCP,soc#7), remote capabilities:
11:25:39 NOTI Capabilities-Exchange-Request(257)[R--], Length=200, Hop-By-Ho
nair4G.eur" }, { Origin-State-Id(278)[-M]=1607973939 (0x5fd7bc33) }, { Host-IP-Addr
-R revision(267)[--]=10201 (0x27d9) }, { Inband-Security-Id(299)[-M]='NO_INBAND_SECUR
{ Vendor-Id(266)[-M]=10415 (0x28af) } }, { Supported-Vendor-Id(265)[-M]=10415 (0x2
11:25:39 DBG SENT to 'epc.openair4G.eur': 'Capabilities-Exchange-Answer'0/257 f
,C:260/l:32,C:265/l:12}
11:25:39 NOTI No TLS protection negotiated with peer 'epc.openair4G.eur'.
11:25:39 NOTI 'STATE_CLOSED' -> 'STATE_OPEN' 'epc.openair4G.eur'

```

Figura 4.4.: Resultat de l'execució de l'MME a l'HSS.

Configuració del SPGW

Igual que als anteriors mòduls, el primer pas és aconseguir l'arxiu de configuració amb la comanda:

```
sudo oai-cn.spgw-conf-get
```

Aquest cop només s'edita l'arxiu `/var/snap/oai-cn/current/spgw.conf`, on s'editen els següents paràmetres principals:

- `SGW_INTERFACE_NAME_FOR_S11`: especifica el nom de la interfície de xarxa que utilitza l'MME per a simular la interfície S11-MME de LTE que connecta l'MME amb el SPGW. Es configura amb `lo`, ja que, els dos mòduls s'executen a la mateixa màquina i, per tant, és la interfície local. Es deixa el valor per defecte.
- `SGW_IPV4_ADDRESS_FOR_S11`: especifica la direcció IP utilitzada pel SPGW per la interfície S11-MME. Ha de coincidir amb el fitxer MME. Es configura com `est127.0.11.2/8`.
- `SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP`: especifica el nom de la interfície de xarxa empleat per l'SPGW que simula la interfície de xarxa S1-U que el comunica amb l'eNB. Es configura com `ens33`.
- `SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP`: especifica la direcció IP empleat per l'SPGW de la interfície de xarxa S1-U. Es configura com `192.168.1.160/24`.
- `PGW_INTERFACE_NAME_FOR_SGI`: especifica el nom de la interfície de xarxa empleat per l'SPGW que simula la interfície de xarxa SGI que connecta el sistema amb internet. Es configura com `ens33`.
- `UE_TCP_MSS_CLAMPING`: especifica si el SPGW realitza NAT (*Network Address Translation* o Traducció d'Adreces de Xarxa) pels UEs connectats al prototip. Es configura amb `yes`.

- `IP_ADDRESS_POOL`: especifica el rang d'adreces IP que la xarxa pot donar als UEs que es connectin. Està compost per la llista `IPV4_LIST` i es configura com `172.16.0.0/12`.
- `DEFAULT_DNS_IPV4_ADDRESS` i `DEFAULT_DNS_SEC_IPV4_ADDRESS`: especifiquen quins DNS (*Domain Name System* o Sistema de Noms de Domini) s'assignaran als UEs que es connectin a la xarxa. És configurem com `1.1.1.1` i `8.8.4.4` respectivament.
- Els altres paràmetres es deixen per defecte.

De la mateixa forma que als apartats anteriors, primer es generen els certificats amb la comanda:

```
oai-cn.spgw-init
```

I, seguidament es comprova el seu funcionament:

```
sudo oai-cn.spgw
```

El resultat que s'ha d'observar pel terminal és que les tasques `GTV1U interface` i `SPGW-APP task interface` estiguin a `done` (o fetes). Això es pot observar a la següent figura:

```
111 GTP kernel configured
000147 00000:513310 7F0CF6622740 DEBUG GTPv1- gw/src/src/gtpv1-u/gtpv1u_task.c:0
124 Initializing GTPV1U interface: DONE
000148 00000:701616 7F0CF6622740 DEBUG SPGW-A arts/spgw/src/src/sgw/sgw_task.c:0
208 Initializing SPGW-APP task interface: DONE
```

Figura 4.5.: Resultat de l'execució de l'SPGW.

4.4.2. Configuració de l'eNB

El primer pas per a començar la configuració de l'estació base de la xarxa LTE consisteix a aconseguir els fitxers de configuració. S'usa la comanda:

```
oai-ran.enb-conf-get
```

Per la configuració de l'escenari s'ha de modificar un arxiu predefinit, per aquest motiu, es fa una còpia de l'arxiu `/var/snap/oai-ran/current/enb.band7.tm1.50PRB.usrpb210.conf` que passa a anomenar-se `enb.band7.tm1.25PRB.usrpb210.conf`. Perquè sigui un fitxer vàlid el primer paràmetre que s'ha de modificar és `N_RB_DL` que determina el nom de *Resource Blocks* (o Blocs de Recursos) que es poden assignar a un UE i es canvia de 50 a 25. Amb aquesta configuració s'usa una amplada de banda de 5 MHz.

També s'han de modificar els següents paràmetres per a obtenir una estació base funcional:

- *tracking_area_code*: especifica el codi d'àrea de seguiment que ha de coincidir amb la configuració de l'MME. Es configura com a 1.
- *plmn_list*: especifica la llista plmn (*Public Land Mobile Network* o Xarxa Mòbil Terrestre Pública). Està compost pel MCC, el MNC i *mnc_length* que determina la longitud del MNC. Aquest últim és un paràmetre important a tenir en compte perquè a Estats Units usen codis de MNC de tres dígits. La configuració ha de coincidir amb la de l'MME i és la següent:
 - *MCC*: 208
 - *MNC*: 92
 - *mnc_length*: 2
- *mme_ip_address*: especifica la direcció IP de l'MME. Es configura com 192.168.1.160.
- *NETWORK_INTERFACES*: especifica els paràmetres de xarxa de l'estació base. Està compost per:
 - *ENB_INTERFACE_NAME_FOR_S1_MME*: especifica el nom de la xarxa empleada per l'eNB per a simular la interfície S1-MME. Es configura com *wlp2s0*.
 - *ENB_IPV4_ADDRESS_FOR_S1_MME*: especifica la direcció IP de la xarxa empleada per l'eNB per a la interfície S1-MME. Es configura com 192.168.1.150/24.
 - *ENB_INTERFACE_NAME_FOR_S1U*: especifica el nom de la xarxa empleada per l'eNB per a simular la interfície S1U. Serveix per a comunicar-se amb l'SPGW. Es configura com *wlp2s0*.
 - *ENB_IPV4_ADDRESS_FOR_S1U*: especifica la direcció IP de la xarxa empleada per l'eNB per a la interfície S1U. És la IP que utilitza l'eNB per a comunicar-se amb l'SPGW. Es configura com 192.168.1.150/24.
- *NETWORK_CONTROLLER*: especifica si s'usa un controlador de la xarxa (*Flex-RAN*). De moment, es configura el paràmetre *FLEXRAN_ENABLED* a *no*.
- Els altres paràmetres es deixen per defecte.

Un cop configurat, es verifica el seu funcionament amb la comanda:

```
sudo oai-ran.enb
```

Pel terminal s'ha d'observar alguna cosa d'aquest estil:

```
[PHY] prach_I0 = 0.1 dB
[PHY] max_I0 32, min_I0 24
[PHY] prach_I0 = 0.1 dB
[PHY] max_I0 32, min_I0 24
[PHY] prach_I0 = 0.1 dB
[PHY] max_I0 32, min_I0 27
```

Figura 4.6.: Resultat de l'execució de l'eNB.

Si es tenen els tres mòduls de l'EPC funcionant, al terminal de l'MME s'ha de poder veure que la taula ha canviat respecte a la figura 4.3:

```
===== STATISTICS =====
Connected eNBs | Current Status | Added since last display | Removed since last display |
Attached UEs   |                |                          |                             |
Connected UEs  |                |                          |                             |
Default Bearers |                |                          |                             |
S1-U Bearers   |                |                          |                             |
```

Figura 4.7.: Resultat de l'execució de l'eNB al MME.

El resultat de la figura 4.7 indica que el nucli de la xarxa, concretament l'MME, ha detectat que l'estació base eNB s'ha connectat correctament a l'EPC. Això es pot observar al paràmetre *Connected eNBs*. Aquest fet suposa un pas endavant per la creació del prototip.

4.4.3. Configuració i gestió d'usuaris de l'HSS

Un cop configurats els dos principals elements del prototip s'han d'afegir els usuaris de la xarxa a la base de dades que prèviament s'ha instal·lat.

Hi ha dues opcions per la gestió d'usuaris. Es pot fer tot des d'una interfície gràfica a *localhost* instal·lant *phpmyadmin* o des de la terminal amb línies d'ordres de MySQL. Al projecte s'ha optat per usar la segona opció per la familiaritat que s'ha obtingut amb el llenguatge que utilitza la base de dades durant el grau.

El primer pas consisteix a accedir a la base de dades. S'usa la comanda:

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

Demana una contrasenya que és la que s'ha configurat amb la instal·lació de *mysql-server*, és a dir, *linux*.

Una vegada que s'ha accedit s'ha d'especificar quina base de dades es vol editar, en aquest cas, *oai_db*. Aquest pas es fa amb la comanda:

```
use oai_db;
```


A continuació, es disposa a editar les següents taules: *users*, *mmeidentity* i *pdn*. Cal destacar que totes les dades afegides s'han obtingut de la base de dades que *Open Cells* cedeix i que com no hi ha forma de saber exactament que SIM es disposa al projecte fins que l'UE no es connecti per primer cop a la xarxa, primerament s'afegeixen tots els usuaris possibles d'*Open Cells* i després s'eliminen tots els que no siguin necessaris. Als següents apartats es veu el procediment amb l'usuari que usa el prototip per a simplificar-ho.

Taula *users*

En aquesta taula s'especifica els usuaris que utilitzen la xarxa que a aquest projecte només és un. S'ha d'especificar que a aquesta taula no s'ha modificat cap paràmetre i que s'usa íntegrament les dades d'*Open Cells* sense cap mena de canvi. Els paràmetres més importants a destacar són:

- **IMSI**: especifica el *International Mobile Subscriber Identity* (o Identitat de l'Abonat mòbil Internacional) que és un codi únic per a cada dispositiu de telefonia mòbil integrat a la targeta SIM. Es deixa per defecte *208920100001102* perquè la SIM que es disposa al projecte ve configurada d'aquesta forma.
- **IMEI**: especifica un codi únic internacional per a la connexió de les xarxes mòbils. Es deixa el valor per defecte.
- **KEY**: especifica la clau d'autenticació de la tarjeta SIM, està formada per 32 caràcters en hexadecimal. Es deixa per defecte, *0x6874736969202073796D4B2079650A73*
- **OPc**: especifica el resultat d'aplicar una operació criptogràfica a **KEY**. És una clau per la protecció de la identitat de les SIMs que és única per a cada una d'elles. D'aquesta forma, si es descobreix la clau, només es pot falsificar una SIM i no totes. Per a obtenir la clau es fa la següent operació:

$$OPc = (KEY \text{RijndaelEncrypt}(OP)) \oplus OP$$

On *RijndaelEncrypt* és una operació de xifrat simètrica i *OP* és la clau que s'ha definit a l'HSS que al projecte està en blanc. D'aquesta forma la SIM imposa la seva clau i no s'ha de realitzar el càlcul ni cal modificar les dades de l'usuari. Finalment, es deixa per defecte, *0x504F20634F6320504F50206363500A4F*

La comanda per afegir a l'usuari és:

```
INSERT INTO 'users'
('imsi','msisdn','imei','imei_sv','ms_ps_status','rau_tau_timer','ue_ambr_ul',
'ue_ambr_dl','access_restriction','mme_cap','mmeidentity_idmmeidentity', 'key','RFSP-Index',
'urrrp_mme', 'sqn', 'rand', 'OPc') VALUES
('208920100001102','33638020002','35609204079202',NULL,'PURGED',120,50000000,100000000,47,
0000000000,1,0x6874736969202073796D4B2079650A73,1,0,000000000000000000351,0x00,
0x504F20634F6320504F50206363500A4F);
```

Taula *mmeidentity*

En aquesta taula s'especifiquen els MMEs del protòtip. En aquest cas sí que s'editen les dades, ja que, el MME del projecte té un nom concret i no s'usen unes dades predeterminades com abans. Els paràmetres més importants a destacar són:

- *idmmeidentity*: especifica l'identificador de l'MME. Com ja hi ha sis MMEs afegits es configura com 7.
- *mmehost*: especifica l'FQDN de l'MME. Es configura com *epc.openair4G.eur*.
- *mmerealm*: especifica l'àmbit de l'MME. Es configura com *openair4G.eur*
- *UE-Reachability*: especifica el valor inicial dels UEs que ha detectat. S'inicialitza a 0.

La comanda per afegir l'MME del projecte és:

```
INSERT INTO 'mmeidentity'  
( 'idmmeidentity', 'mmehost', 'mmerealm', 'UE-Reachability' ) VALUES  
(7, 'epc.openair4G.eur', 'openair4G.eur', 0);
```

Taula *pdn*

En aquesta taula especifica la informació dels usuaris i l'associació a un APN (*Access Point Name* o Nom de Punt d'Accés). En aquest cas s'edita només el camp d'APN on determina l'APN que usa l'UE. Els paràmetres més importants a destacar són:

- *id*: especifica l'identificador del perfil de l'usuari. Es deixa per defecte, 24.
- *apn*: especifica l'APN que utilitza l'UE. Es configura com *oai.ipv4*.
- *pdn_type*: especifica el tipus d'APN. Es deixa per defecte, *IPv4*.
- *user_imsi*: especifica l'IMSI de la targeta SIM vinculada. Es deixa per defecte *208920100001102*.

La comanda per afegir l'APN del projecte és:

```
INSERT INTO 'pdn'  
( 'id', 'apn', 'pdn_type', 'pdn_ipv4', 'pdn_ipv6', 'aggregate_ambr_ul', 'aggregate_ambr_dl',  
'pgw_id', 'users_imsi', 'qci', 'priority_level', 'pre_emp_cap', 'pre_emp_vul', 'LIPA-Permissions' )  
VALUES (24, 'oai.ipv4', 'IPv4', '0.0.0.0', '0:0:0:0:0:0', 50000000, 100000000, 3,  
'208920100001102', 9, 15, 'DISABLED', 'ENABLED', 'LIPA-only');
```

4.4.4. Configuració del dongle

Per a configurar el *dongle* s'accedeix a la interfície gràfica a partir del navegador *Firefox*. Per accedir-hi només s'ha d'anar a la IP: 192.168.8.1 que és la IP que defineix el *dongle* per defecte. Es visualitza la següent pantalla:

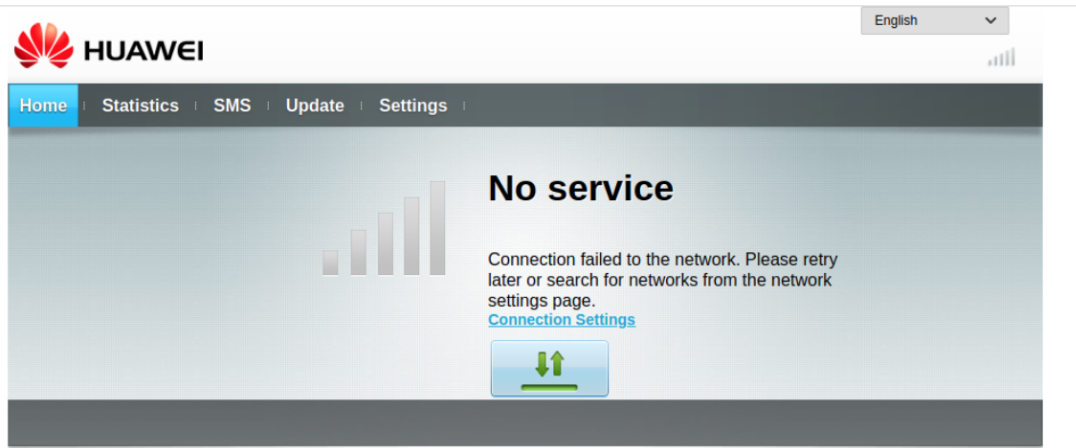


Figura 4.8.: Interfície gràfica del dongle.

A continuació, es va a la pestanya *Settings* (o configuració), concretament, a l'apartat de *Profile Management* (o Gestió de Perfils). Una vegada en aquest apartat s'edita el camp APN que ha de coincidir amb l'especificat a la taula PDN de la base de dades. Per tant, es configura com *oai.ipv4*. Tal com es veu a la següent figura:

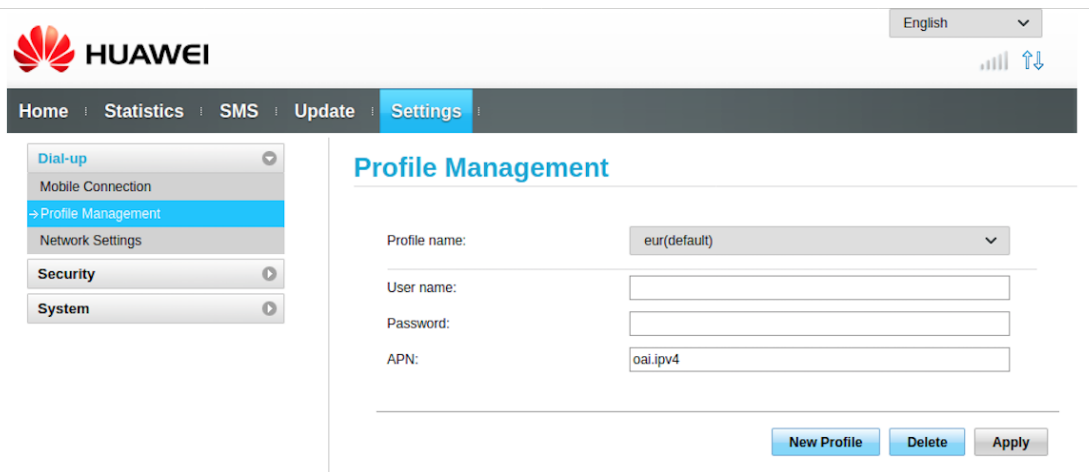


Figura 4.9.: Configuració del dongle.

Per la verificació de la configuració de la base de dades i del *dongle* s'ha d'iniciar el nucli de xarxa (EPC) amb les següents instruccions cada una en diferents terminals i en

aquest ordre:

```
sudo oai-cn.hss
sudo oai-cn.mme
sudo oai-cn.spgw
```

I, a continuació, s'inicia la xarxa d'accés de radi (l'eNB) amb la comanda:

```
sudo oai-ran.enb
```

El prototip hauria de ser capaç de detectar l'UE, és a dir, el *dongle* connectat a la *Raspberry*. Això es pot veure des de l'MME a la següent imatge:

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

Figura 4.10.: Verificació de la connexió de l'UE amb la xarxa LTE.

A la figura 4.10 es pot apreciar com el nucli de xarxa ha sigut capaç de connectar-se tant amb l'eNB mitjançant la interfície de xarxa S1-U com amb l'UE.

Pel que fa al *dongle* s'ha d'apreciar la següent pantalla:

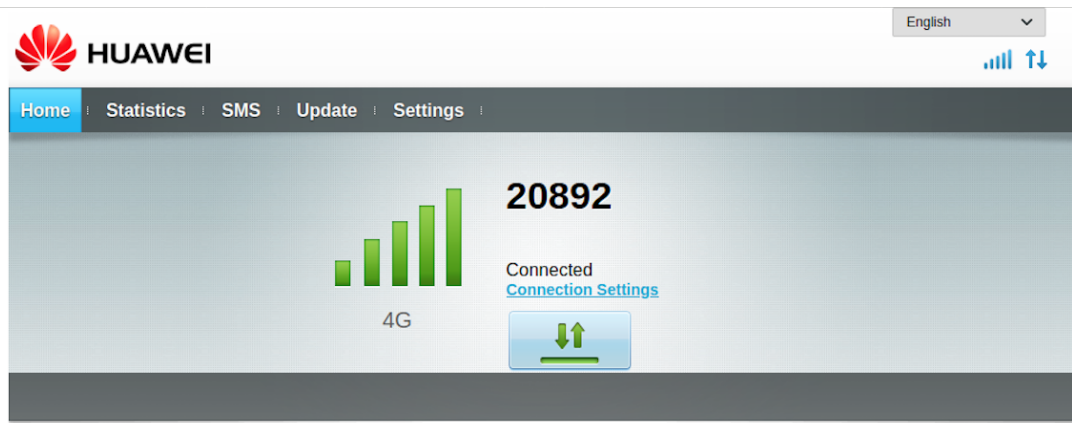


Figura 4.11.: Verificació de la connexió de l'UE al dongle.

El dongle dona diferent informació sobre l'UE. Informa sobre l'MCC (208) i l'MNC (92) i la cobertura a temps real que té la *Raspberry*. En aquest punt, el prototip, ja és capaç de connectar l'UE a internet.

4.4.5. Configuració del *FlexRAN*

Mosaic5G presenta una configuració molt senzilla del *FlexRAN*. El primer pas consisteix a recuperar l'arxiu de configuració de l'eNB, és a dir, *enb.band7.tm1.25PRB.usrbp210.conf* i anar al camp *NETWORK_CONTROLLER* (o Controlador de la Xarxa) on es troben els següents paràmetres més destacats:

- *FLEXRAN_ENABLED*: especifica si es vol que *FlexRAN* monitori la xarxa LTE. Es configura com a *yes*.
- *FLEXRAN_INTERFACE_NAME*: especifica la interfície de xarxa on actua *FlexRAN*. Com s'executa a la mateixa màquina que l'eNB es configura com *lo*, és a dir, *localhost*.
- *FLEXRAN_IPV4_ADDRESS*: especifica l'adreça IP que utilitza *FlexRAN* per a monitorar la xarxa LTE. Es configura com *127.0.0.1*.
- Els altres paràmetres es deixen per defecte.

Per a comprovar el funcionament del *FlexRAN* s'executa la comanda:

```
sudo flexran
```

Pel terminal s'ha de veure:

```
diego@eNB:~$ sudo flexran
/snap/flexran/25/bin/flexran -p 2210 -n 9999 -s 0.0.0.0 -a 0.0.0.0
[INFO][FLEXRAN_RTC] - Listening on 0.0.0.0:2210 for incoming agent connections
[INFO][FLEXRAN_RTC] - Listening on 0.0.0.0:9999 for incoming REST connections
```

Figura 4.12.: Verificació del funcionament del *FlexRAN*.

En aquest punt el *FlexRAN* està esperant la connexió de l'eNB i de l'UE. Una vegada que està activat es poden veure algunes dades del prototip pel terminal. Per exemple notifica si una estació base s'ha connectat a la xarxa i informa del seu identificador. Passa el mateix quan es connecta un UE notifica de la connexió i del seu RNTI (*Radio Network Temporary Identifier* o Identificador Temporal de la Xarxa de Radi). Per a veure més detalls s'ha d'anar al navegador i buscar *localhost:9999/stats* tal com es veurà a la secció cinc.

4.4.6. Configuració del *Slicing*

Finalment, la configuració del sistema finalitza amb l'*slicing*. Aquest últim pas es durà a terme mitjançant peticions *http* al *FlexRAN* amb la comanda *curl*. Les peticions estan compostes primerament per la direcció *url* del *FlexRAN* i seguidament per un arxiu *json* amb les dades de les *slices*.

Per exemple, si es volen crear *slices*, es du a terme la comanda:

```
curl -X POST http://127.0.0.1:9999/slice/enb/-1 --data-binary @ran-sharing.json
```

En aquest cas, l'arxiu *ran-sharing.json* crea dues *slices* que comparteixen els mateixos recursos de l'espectre. En el cas que es vol crear més *slices* o canviar la repartició dels recursos s'ha de tornar a editar el fitxer i fer, un altre cop, la comanda anterior.

La url *http://127.0.0.1:9999/slice/enb/-1* indica el lloc on es creen les *slices*. Que, amb el *-1*, es formen a l'últim eNB que s'ha connectat a l'EPC i que *FlexRAN* ha detectat.

I, per associar un UE a qualsevol de les dues *slices*, s'usa la comanda:

```
curl -X POST http://127.0.0.1:9999/ue_slice_assoc/enb/ --data-binary @ue-association.json
```

El funcionament de la comanda és el mateix que a l'anterior. La *url* indica el lloc on l'UE és associat, en aquest cas, com només tenim una estació base (eNB), ja és suficient indicant que volem que l'UE sigui associat a aquesta.

I el fitxer *ue-association.json* determina a quina *slice* es connecta la *Raspberry*. Si es vol canviar la *slice* associada, s'ha de modificar el fitxer i tornar a fer la comanda d'abans.

Les funcionalitats de les *slices* es veuran en profunditat a la següent secció. El contingut dels fitxers es poden consultar a l'ANNEX C.

5. Resultats

En aquest capítol s'aborden els resultats obtinguts després de l'avaluació de la xarxa LTE enfront de les proves realitzades. L'objectiu d'aquest capítol és verificar la validesa del prototip i confirmar si s'han complert els objectius proposats al projecte.

5.1. Captures de xarxa

En aquest apartat es capturen els missatges de control i senyalització intercanviats entre l'eNB i l'EPC i el tràfic de paquets que crea l'UE quan es connecta a internet i accedeix a determinats serveis. Cal recordar que la IP de l'eNB és 192.168.1.150 i la de l'EPC és 192.168.1.160.

Per a veure els paquets enviats al plànol de control entre l'eNB i l'EPC s'estudia el protocol de xarxa S1 i per a trobar els paquets enviats al plànol d'usuari s'observa el protocol de xarxa GTP.

Per a la realització de la prova s'han de seguir els següents passos:

- Iniciar *wireshark* a un terminal de l'EPC i capturar la interfície de xarxa *ens33*.
- Executar els tres mòduls de software de l'EPC com s'ha vist a l'apartat 4.4.4.
- Posar en marxa l'eNB de la mateixa forma que s'ha vist a l'apartat 4.4.4.
- Esperar fins que l'MME detecti a l'eNB i l'UE.
- Realitzar una cerca al navegador *Firefox* des de la *Raspberry*.

A continuació, s'estudia el protocol de xarxa S1 i després el protocol GTP.

5.1.1. Plànol de control

En aquesta captura es pot observar tot el tràfic intercanviat pel protocol de xarxa S1 implementat per OAI al pla de control que correspon a la senyalització de la interfície S1-MME.

A escala d'aplicació, el tràfic de la senyalització (interfície S1-MME), utilitza el protocol S1AP que es pot observar a la figura 5.1. A aquesta captura es pot veure la informació NAS intercanviada entre l'MME i l'eNB corresponent a diferents procediments de registre, autenticació, activació de contexts, i desactivació que s'expliquen amb més detall a continuació.

El primer missatge intercanviat és una petició d'establiment que l'eNB envia a l'EPC, és a dir, envia una *S1SetupRequest* i l'EPC, via MME, accepta la petició de connexió (*S1SetupResponse*) i els dos elements queden connectats.

Un cop finalitzada la connexió entre l'eNB i l'EPC, l'UE envia un *Attach Request* on informa de la seva identitat a l'MME que el reconeix pel seu IMSI. Un cop que l'MME rep aquest missatge comença tot el procés d'autenticació i seguretat. L'MME demana un vector d'autenticació a l'HSS que, aquest genera utilitzant l'algoritme *EPS AKA*, i ho envia a l'MME. A continuació, l'MME remet aquest vector cap l'eNB i aquest cap l'UE, és a dir, fa una *Authentication request*. Quan l'UE ha obtingut aquest vector el torna a calcular amb l'algoritme *EPS AKA* i, si coincideix amb el vector que li hi ha arribat, ho envia cap a l'estació base i aquesta s'encarrega d'enviar-ho a l'MME, és a dir, fa una *Authentication response*. En aquest punt el protocol d'autenticació ha finalitzat. Finalment, amb el fi de què l'UE activi la seguretat NAS, l'MME envia a l'UE (per mitjà de l'eNB) l'ordre *Security mode command* i l'UE el respon amb *Security mode complete*.

No.	Time	Source	Destination	Protocol	Length	Info
41 27	.049269414	192.168.1.150	192.168.1.160	S1AP	122	S1SetupRequest
43 27	.049499060	192.168.1.160	192.168.1.150	S1AP	90	S1SetupResponse
79 69	.741050907	192.168.1.150	192.168.1.160	S1AP/N...	222	InitialUEMessage, Attach request, PDN connectivity request
80 69	.741984003	192.168.1.160	192.168.1.150	S1AP/N...	110	DownlinkNASTransport, Identity request
81 69	.809986558	192.168.1.150	192.168.1.160	S1AP/N...	138	UplinkNASTransport, Identity response
82 69	.813123001	192.168.1.160	192.168.1.150	S1AP/N...	142	DownlinkNASTransport, Authentication request
83 69	.998955839	192.168.1.150	192.168.1.160	S1AP/N...	138	UplinkNASTransport, Authentication response
84 70	.000105802	192.168.1.160	192.168.1.150	S1AP/N...	122	DownlinkNASTransport, Security mode command
86 70	.049976735	192.168.1.150	192.168.1.160	S1AP/N...	134	UplinkNASTransport, Security mode complete
87 70	.062757411	192.168.1.160	192.168.1.150	S1AP/N...	274	InitialContextSetupRequest, Attach accept, Activate default E...
88 70	.130125811	192.168.1.150	192.168.1.160	S1AP	150	UECapabilityInfoIndication, UECapabilityInformation
91 70	.327596179	192.168.1.150	192.168.1.160	S1AP/N...	182	InitialContextSetupResponse, UplinkNASTransport, Attach compl...

Figura 5.1.: Tràfic de paquets al protocol de xarxa S1AP.

Quan els processos d'autenticació i de seguretat han acabat l'UE s'ha connectat correctament a la xarxa LTE.

5.1.2. Plànol d'usuari

En aquesta captura es pot observar el pla d'usuari (interfície S1-U) que encapsula els paquets de dades de l'usuari mitjançant el protocol GTP. Una part de l'intercanvi de paquets es pot veure a la figura 5.2.

Tal com es pot observar a aquesta la figura, l'UE amb l'IP 172.16.0.2 proporcionada per l'SPGW, ha sigut capaç de connectar-se a internet utilitzant com a via la interfície SGi. Primerament es connecta amb el DNS *1.1.1.1* que l'encamina cap a la direcció 34.107.221.82 que és una organització web de *Google* que serveix per a allotjar webs. Que aquesta li respon amb el protocol de web *http* i aconsegueix arribar a l'IP *147.83.194.33*. És a dir, l'UE està al portal web *Atenea*.

No.	Time	Source	Destination	Protocol	Length	Info
734	79.681842271	34.107.221.82	172.16.0.2	GTP <TCP>	10288	-- 35454 [ACK] Seq=1 Ack=318 Win=295184 Len=0 TSval=1191976...
736	79.682582191	52.89.213.25	172.16.0.2	GTP <TCP>	102443	-- 43934 [ACK] Seq=1 Ack=214 Win=79872 Len=0 TSval=3616232...
738	79.683275200	1.1.1.1	172.16.0.2	GTP <DNS>	187	Standard query response 0xe89f AAAA mozilla.org SOA infoblob1...
740	79.683678750	1.1.1.1	172.16.0.2	GTP <DNS>	168	Standard query response 0x6076 No such name PTR 2.0.a.4.5.f.9...
742	79.683838337	1.1.1.1	172.16.0.2	GTP <DNS>	168	Standard query response 0x10bf No such name PTR b.4.8.c.6.c.a...
744	79.690627163	34.107.221.82	172.16.0.2	GTP <HTTP>	322	HTTP/1.1 200 OK (text/plain)
746	79.691772605	34.107.221.82	172.16.0.2	GTP <HTTP>	322	HTTP/1.1 200 OK (text/plain)
747	79.707109990	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=6101 Win=64128 Len=0 TSval=217...
749	79.734054792	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=7513 Win=64128 Len=0 TSval=217...
752	79.749430478	52.89.213.25	172.16.0.2	GTP <TLSV1.2>	1390	Server Hello
754	79.749533876	52.89.213.25	172.16.0.2	GTP <TCP>	1390	453 -- 43930 [ACK] Seq=1289 Ack=214 Win=83456 Len=1288 TSval=3...
755	79.749556933	52.89.213.25	172.16.0.2	GTP <TLSV1.2>	927	Certificate, Server Key Exchange, Server Hello Done
756	79.761068186	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=8925 Win=63360 Len=0 TSval=217...
758	79.769188750	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=10337 Win=64128 Len=0 TSval=21...
759	79.769200035	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=11749 Win=64128 Len=0 TSval=21...
760	79.769201265	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=13161 Win=64128 Len=0 TSval=21...
761	79.769202308	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=14573 Win=64128 Len=0 TSval=21...
762	79.769203240	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=15985 Win=64128 Len=0 TSval=21...
763	79.769226132	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=17397 Win=64128 Len=0 TSval=21...
764	79.769230491	172.16.0.2	147.83.194.33	GTP <TCP>	10244216	-- 443 [ACK] Seq=1132 Ack=18007 Win=64128 Len=0 TSval=21...
765	79.769231633	172.16.0.2	8.8.4.4	GTP <DNS>	107	Standard query 0xe89f AAAA mozilla.org
766	79.769232580	172.16.0.2	8.8.4.4	GTP <DNS>	168	Standard query 0x6076 PTR 2.0.a.4.5.f.9.7.9.b.8.a.d.0.5.0.0.0...
767	79.769249656	172.16.0.2	8.8.4.4	GTP <DNS>	168	Standard query 0x10bf PTR b.4.8.c.6.c.a.7.b.f.f.4.6.4.6.4.e.9.0.0...

Figura 5.2.: Tràfic de paquets al protocol de xarxa GTP.

5.2. Proves de rendiment, monitoratge i segmentació

Un cop que s'ha confirmat la correcta comunicació entre els diferents mòduls que conformen la xarxa i de determinar que l'UE és capaç de connectar-se a internet és el moment de comprovar el rendiment en diferents escenaris i de monitorar el prototip.

Primer s'explica breument que és *FlexRAN* i perquè s'utilitza al projecte, seguidament es realitzen les proves de rendiment que es fan quan l'UE té la cobertura màxima, és a dir, 5 ratlles i seguint el següent procediment:

- Executar tots els mòduls de software de la mateixa forma que a les proves de *wireshark* amb la diferència que a la màquina eNB també s'executa el *FlexRAN* com s'ha vist a l'apartat 4.4.5.
- Comprovar per comandes de terminal o via web (*localhost:9999/stats*) l'estat actual de la xarxa amb *FlexRAN*.
- Crear amb més o menys recursos les *slices* via comandes de terminal (si cal).
- Assignar a l'UE a una *slice* específica per comandes de terminal (si cal).
- Comprovar el rendiment de la xarxa amb l'eina *iPerf3*.

5.2.1. Funcionament de *FlexRAN*

En aquest apartat s'explica com funciona i com es pot usar l'eina de monitoratge *FlexRAN*.

Una vegada que tots els mòduls de software estan actius es pot consultar l'estat de la xarxa amb *FlexRAN*. Si es vol usar la terminal, existeixen diferents comandes per a obtenir les dades de la xarxa, per exemple, per tenir les dades de l'eNB es fa servir la comanda:

```
curl -X GET http://127.0.0.1:9999/stats/enb/-1/ | jq . | tee ~/Escritorio/flex/FLEXENB
```

La instrucció fa una petició *GET* a la *url* *127.0.0.1:9999/stats/enb/-1/*. Un cop que es té la informació es passa pel programa *jq* que s'encarrega de mantenir el mateix format que hi ha a la pàgina web. Finalment, guardem aquesta informació amb la comanda *tee* a l'arxiu *FLEXENB*.

Per a aconseguir la informació de l'UE també es pot usar la comanda:

```
curl -X GET http://127.0.0.1:9999/stats/ue/208920100001102 | jq . | tee
~/Escritorio/flex/FLEXENB
```

El funcionament de la comanda és exactament el mateix que l'anterior on *208920100001102* és l'IMSI de l'UE.

I, si es vol obtenir la informació completa de la xarxa, s'usa la comanda:

```
curl -X GET http://127.0.0.1:9999/stats/ | jq . | tee ~/Escritorio/flex/FLEX
```

Al projecte s'ha decidit no usar aquest mètode perquè és molt poc visual. Així que es consulta directament la *url* de la comanda (*http://127.0.0.1:9999/stats/*) i, d'aquesta forma, les dades es veuen d'una forma molt més agradable i comprensible.

Per exemple, quan l'eNB està actiu, per la web es pot observar els següents paràmetres:

- L'identificador de l'eNB, que en aquest cas és 10000:

```

▼ eNB_config:
  ▼ 0:
    bs_id: 10000

```

Figura 5.3.: Identificació de l'eNB.

- La configuració de l'MCC (208), l'MNC (92) i de les *slices* que com encara no s'han creat és inexistent. Aquesta informació es pot consultar a la figura 5.4.
- Les dades del protocol de xarxa S1AP per la comunicació amb l'MME, la IP i l'estat d'aquest. Aquesta informació es pot consultar a la figura 5.5.

- La banda ampla de l'eNB en Resource Blocks que determina els recursos màxims per a les *slices*. Aquesta informació es pot consultar a la figura 5.6.

```

▼ plmnId:
  ▼ 0:
    mcc: 208
    mnc: 92
    mnclength: 2
  ▼ sliceConfig:
    ▼ dl:
      algorithm: "None"
      scheduler: "round_robin_dl"
    ▼ ul:
      algorithm: "None"
      scheduler: "round_robin_ul"

```

Figura 5.4.: Configuració de l'MCC, l'MNC i de les slices.

```

▼ slap:
  pending: 0
  connected: 1
  enbSlIp: "192.168.1.150"
  enbName: "eNB-Eurecom-LTEBox"
  ▼ mme:
    ▼ 0:
      slIp: "192.168.1.160"
      state: "FLMMES_CONNECTED"

```

Figura 5.5.: Configuració del protocol S1AP i de l'MME.

```

dlBandwidth: 25
ulBandwidth: 25

```

Figura 5.6.: Nombre de *Resource Blocks*.

Aquestes dades són les més importants de l'eNB per a la monitorització del projecte, no obstant això *FlexRAN* dona més informació sobre l'estat de l'eNB. Un cop que la *Raspberry* es connecta a la xarxa les dades de l'estació base s'actualitzen i apareix l'apartat de l'UE on mostra la informació de l'usuari de la xarxa. Aquest són els paràmetres que es tindran en compte:

- El RNTI que en aquest cas és 54185. Aquesta informació es pot consultar a la figura 5.7.
- L'*IMSI* de l'*UE* que és 208920100001102.

- La configuració de l'*MCC* (208), l'*MNC* (92) i del protocol de xarxa *S1AP* que és la mateixa que a l'*eNB*.
- La configuració de les *slices* que com no s'han creat encara és inexistent.

rnti: 54185

Figura 5.7.: *RNTI* de l'*UE*.

Per tant, per a tots els diferents escenaris que es plantegen a continuació, es consultarà *FlexRAN* per determinar l'estat de la xarxa i la correcta creació de les *slices* i, d'aquesta forma, demostrar visualment el funcionament exitós del prototip i dels objectius proposats. No obstant això, només s'adjuntaran captures de pantalla al segon escenari per evitar la repetició excessiva als procediments.

5.2.2. Proves de rendiment

En aquest apartat s'estudia el rendiment de la xarxa per a quatre escenaris hipotètics:

- Primer escenari: l'*UE* gaudeix del 100% dels recursos de la xarxa tant pel *uplink* (o enllaç de pujada) com pel *downlink* (o enllaç de baixada), és a dir, no es divideix l'espectre, no es creen les *slices*.
- Segon escenari: l'*UE* gaudeix del 50% dels recursos de la xarxa tant pel *uplink* com pel *downlink*, és a dir, es creen dos *slices* que comparteixen a parts iguals l'espectre.
- Tercer escenari: l'*UE* gaudeix del 75% dels recursos de la xarxa pel *uplink* i del 66% dels recursos pel *downlink*, és a dir, es creen dos *slices* on una d'elles utilitza la majoria de l'espectre.
- Quart escenari: l'*UE* gaudeix del 25% dels recursos de la xarxa pel *uplink* i del 33% dels recursos pel *downlink*, és a dir, es creen dos *slices* on una d'elles utilitza de manera molt reduïda l'espectre.

Com ja es va explicar a l'apartat 4.4.2 i com ja s'ha comprovat per mitjà de *FlexRAN* a l'apartat 5.2.1 el nombre màxim de *Resource Blocks* configurat al prototip és de 25. Per tant, en tots els escenaris, es creen dos *slices* modificant la quantitat de *Resource Blocks* que se li assigna a cada *slice* i així aconseguir els diferents percentatges proposats.

Primer escenari

En aquest primer escenari només s'han d'iniciar tots els mòduls de *software* tal com s'ha explicat a l'apartat 5.2 i no cal crear cap *slice*. Així s'aconsegueix obtenir el rendiment màxim que pot oferir el prototip. Quan l'*UE* estigui connectat a la xarxa s'ha d'obrir un terminal a l'EPC i a la *Raspberry*. L'EPC fa de servidor i la *Raspberry* de client. Per a dur a terme aquest montatge s'utilitza l'eina *iPerf3*.

Al terminal de l'EPC s'usa la comanda:

```
iperf3 -s
```

Amb el `-s` s'aconsegueix que l'EPC tingui el paper de servidor. I a la *Raspberry* la comanda que s'utilitza perquè sigui un client és:

```
iperf3 -c 192.168.1.160
```

On `-c` indica que és el client i que s'ha de connectar al servidor amb la IP 192.168.1.160, és a dir, a l'EPC.

Amb aquest muntatge s'obté el rendiment pel canal d'*uplink* que fa referència a la velocitat màxima que es pot oferir quan l'UE es comunica amb l'EPC. Els resultats es veuen des de la part del servidor, per tant, s'ha de tenir en compte el valor de *receiver*, ja que, en aquest cas ha estat el servidor el que rebia les dades. El resultat és el següent:

```
diego@epc:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 172.16.0.2, port 35744
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 35746
[ ID] Interval      Transfer    Bandwidth
[ 5]  0.00-1.00    sec        592 KBytes  4.85 Mbits/sec
[ 5]  1.00-2.00    sec        909 KBytes  7.45 Mbits/sec
[ 5]  2.00-3.00    sec        946 KBytes  7.75 Mbits/sec
[ 5]  3.00-4.00    sec        946 KBytes  7.75 Mbits/sec
[ 5]  4.00-5.00    sec        947 KBytes  7.76 Mbits/sec
[ 5]  5.00-6.00    sec        946 KBytes  7.75 Mbits/sec
[ 5]  6.00-7.00    sec        947 KBytes  7.76 Mbits/sec
[ 5]  7.00-8.00    sec        946 KBytes  7.75 Mbits/sec
[ 5]  8.00-9.00    sec        946 KBytes  7.75 Mbits/sec
[ 5]  9.00-10.00   sec        947 KBytes  7.76 Mbits/sec
[ 5] 10.00-10.55   sec        522 KBytes  7.75 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5]  0.00-10.55   sec       11.5 MBytes  9.11 Mbits/sec    0
[ 5]  0.00-10.55   sec         9.37 MBytes  7.45 Mbits/sec
-----
sender
receiver
```

Figura 5.8.: *Uplink* al 100%.

Tal com es veu a la figura 5.8 la velocitat d'*uplink* és de 7.45Mbits/sec dins del límit teòric que pot proporcionar *OAI* que està al voltant de 7.5Mbits/sec .

Per a calcular el *downlink*, que fa referència a la velocitat màxima que es pot oferir quan l'EPC es comunica amb l'UE, el muntatge és exactament igual amb les diferències de què a la terminal de la *Raspberry* s'ha d'executar la comanda:

```
iperf3 -c 192.168.1.160 -R
```

Com s'està calculant el *downlink* s'ha de tenir en compte el valor *sender*, ja que, en aquest cas ha estat el servidor que s'ha comunicat amb l'UE, és a dir, ha enviat dades. El resultat és el següent:

```
Accepted connection from 172.16.0.2, port 35754
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 35756
[ ID] Interval      Transfer      Bandwidth    Retr  Cwnd
[ 5]  0.00-1.00    sec  826 KBytes   6.76 Mbits/sec  0    83.4 KBytes
[ 5]  1.00-2.00    sec  1.29 MBytes  10.8 Mbits/sec  0    143 KBytes
[ 5]  2.00-3.00    sec  1.68 MBytes  14.1 Mbits/sec  0    229 KBytes
[ 5]  3.00-4.00    sec  2.31 MBytes  19.4 Mbits/sec  0    335 KBytes
[ 5]  4.00-5.00    sec  2.12 MBytes  17.8 Mbits/sec  0    441 KBytes
[ 5]  5.00-6.00    sec  2.09 MBytes  17.6 Mbits/sec  0    543 KBytes
[ 5]  6.00-7.00    sec  2.33 MBytes  19.5 Mbits/sec  0    648 KBytes
[ 5]  7.00-8.00    sec  2.10 MBytes  17.6 Mbits/sec  0    751 KBytes
[ 5]  8.00-9.00    sec  2.12 MBytes  17.8 Mbits/sec  0    861 KBytes
[ 5]  9.00-10.00   sec  1.93 MBytes  16.2 Mbits/sec  0    963 KBytes
[ 5] 10.00-10.03   sec  0.00 Bytes   0.00 bits/sec  0    966 KBytes
-----
[ ID] Interval      Transfer      Bandwidth    Retr
[ 5]  0.00-10.03   sec  18.8 MBytes  15.7 Mbits/sec  0
[ 5]  0.00-10.03   sec  18.0 MBytes  15.0 Mbits/sec

sender
receiver
```

Figura 5.9.: *Downlink* al 100%.

A la figura 5.9 es veu com la velocitat màxima per *Downlink* és de 15.7Mbits/sec . També dins del límit teòric del màxim que pot proporcionar OAI que és al voltant de 17Mbits/sec . La diferència entre la velocitat de pujada i de baixada és tan gran perquè OAI en l'àmbit d'implementació dona més recursos al *downlink*.

Segon escenari

El procediment a seguir en aquest escenari són exactament els mateixos que a l'escenari anterior amb el pas intermedi de la creació de les *slices* i l'assignació de l'UE a una d'elles. Aquest pas intermedi es realitza de la mateixa forma que s'ha explicat a l'apartat 4.4.6.

Per la creació d'aquestes dues *slices* que comparteixen el 50% de l'espectre s'ha de modificar les posicions dels *Resource Blocks* a l'arxiu *ran-sharing.json*. Per l'*uplink*:

```
{
  "id": 0,
  "label": "default",
  "static": {
    "posLow": 1,
    "posHigh": 12
  }
}
{id": 2,
  "label": "two",
  "static": {
    "posLow": 13,
    "posHigh": 25
  }
}
```

I pel *downlink*:

```
"id": 0,
  "label": "default",
  "static": {
    "posLow": 0,
    "posHigh": 5
  }
}
"id": 2,
  "label": "default",
  "static": {
    "posLow": 6,
    "posHigh": 12
  }
}
```

D'aquesta forma s'han creat dues *slices* amb identificadors 0 i 2 que comparteixen els mateixos recursos. Els paràmetres *posLow* i *posHigh* determinen el nombre de *Resource Blocks* que té cada *slice*, com més diferència hi hagi entre els dos paràmetres més recursos s'assignen a una *slice*. Cal destacar que, les *slices* no podem compartir un *Resource Block* en específic, és a dir, no es pot donar el bloc 12 a les dues *slices*.

I, per assignar l'UE a una *slice*, s'ha d'especificar el seu IMSI i determinar l'identificador de la *slice* (tant de pujada com de baixa) que es vol assignar a l'UE. Per a fer-ho es modifica el fitxer *ue-association.json*. Com que en aquest cas les dues *slices* comparteixen els mateixos recursos, es destina l'*Uplink* de l'*UE* a la *slice* amb identificador 0 i el *Downlink* a la *slice* amb identificador 2. El procediment és el següent:

```
"ueConfig": [
  {
    "imsi": 208920100001102,
    "dlSliceId": 2,
    "ulSliceId": 0
  }
]
```

Tal com es veu al codi, un UE pot ser assignat a diferents *slices* per l'*Uplink* i pel *Downlink*. Això dona molta flexibilitat al prototip de xarxa LTE, en tenir aquesta característica pot assignar i distribuir els recursos que ofereix d'una forma òptima. D'aquesta forma, la xarxa, s'assegura de donar un bon servei als usuaris que es connectin.

Cal destacar que l'assignació a diferents *slices* de pujada i de baixada es pot fer sempre que es vulgui, no és necessari que les *slices* comparteixen exactament la meitat dels recursos de l'espectre.

A continuació, per a comprovar la correcta creació de les *slices* i de la posterior assignació de l'UE, s'ha de consultar la interfície gràfica de *FlexRAN*. Per a fer-ho s'ha d'obrir el navegador *Firefox* i anar a l'adreça web <http://127.0.0.1:9999/stats/>, buscar l'apartat de l'UE i, seguidament, la configuració de les *slices*. Així es pot observar la següent informació:

```

▼ sliceConfig:
  ▼ dl:
    algorithm: "Static"
    ▼ slices:
      ▼ 0:
        id: 0
        label: "default"
        scheduler: "round_robin_dl"
        ▼ static:
          posLow: 0
          posHigh: 5
      ▼ 1:
        id: 2
        label: "two"
        scheduler: "round_robin_dl"
        ▼ static:
          posLow: 6
          posHigh: 12
  ▼ ul:
    algorithm: "Static"
    ▼ slices:
      ▼ 0:
        id: 0
        label: "default"
        scheduler: "round_robin_ul"
        ▼ static:
          posLow: 1
          posHigh: 12
      ▼ 1:
        id: 2
        label: "two"
        scheduler: "round_robin_ul"
        ▼ static:
          posLow: 13
          posHigh: 25

```

Figura 5.10.: Creació de les *slices*.

```

dlSliceId: 2
ulSliceId: 0

```

Figura 5.11.: Assignació de l'UE a les *slices*.

Un cop que s'han creat correctament les *slices* es pot fer el test de rendiment amb *iPerf3* de la mateixa forma que a l'escenari anterior. Per l'*uplink*:

```
diego@epc:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 172.16.0.2, port 60918
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 60922
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-1.00    sec 448 KBytes 3.67 Mbits/sec
[ 5] 1.00-2.00    sec 498 KBytes 4.08 Mbits/sec
[ 5] 2.00-3.00    sec 498 KBytes 4.08 Mbits/sec
[ 5] 3.00-4.00    sec 498 KBytes 4.08 Mbits/sec
[ 5] 4.00-5.00    sec 499 KBytes 4.09 Mbits/sec
[ 5] 5.00-6.00    sec 498 KBytes 4.08 Mbits/sec
[ 5] 6.00-7.00    sec 498 KBytes 4.08 Mbits/sec
[ 5] 7.00-8.00    sec 499 KBytes 4.09 Mbits/sec
[ 5] 8.00-9.00    sec 498 KBytes 4.08 Mbits/sec
[ 5] 9.00-10.00   sec 498 KBytes 4.08 Mbits/sec
[ 5] 10.00-10.97  sec 482 KBytes 4.08 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5] 0.00-10.97   sec 7.00 MBytes 5.36 Mbits/sec 0
[ 5] 0.00-10.97   sec 5.29 MBytes 4.04 Mbits/sec
-----
sender
receiver
```

Figura 5.12.: *Uplink* al 50%.

En aquest cas, la velocitat de pujada és de 4.04Mbits/sec que és, aproximadament, la meitat de la velocitat de l'anterior escenari.

I pel *downlink*:

```
Accepted connection from 172.16.0.2, port 37604
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 37606
[ ID] Interval      Transfer    Bandwidth    Retr  Cwnd
[ 5] 0.00-1.00    sec 370 KBytes 3.03 Mbits/sec 0    29.7 KBytes
[ 5] 1.00-2.00    sec 491 KBytes 4.02 Mbits/sec 0    52.3 KBytes
[ 5] 2.00-3.00    sec 833 KBytes 6.82 Mbits/sec 0    90.5 KBytes
[ 5] 3.00-4.00    sec 1.01 MBytes 8.44 Mbits/sec 0    140 KBytes
[ 5] 4.00-5.00    sec 1.20 MBytes 10.0 Mbits/sec 0    197 KBytes
[ 5] 5.00-6.00    sec 1.07 MBytes 8.97 Mbits/sec 0    249 KBytes
[ 5] 6.00-7.00    sec 1.25 MBytes 10.5 Mbits/sec 0    303 KBytes
[ 5] 7.00-8.00    sec 1.05 MBytes 8.82 Mbits/sec 0    359 KBytes
[ 5] 8.00-9.00    sec 1.17 MBytes 9.80 Mbits/sec 0    413 KBytes
[ 5] 9.00-10.00   sec 1.23 MBytes 10.4 Mbits/sec 0    519 KBytes
[ 5] 10.00-10.05  sec 0.00 Bytes 0.00 bits/sec 0    532 KBytes
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5] 0.00-10.05   sec 9.63 MBytes 8.04 Mbits/sec 0
[ 5] 0.00-10.05   sec 8.92 MBytes 7.44 Mbits/sec
-----
sender
receiver
```

Figura 5.13.: *Downlink* al 50%.

Tenim el mateix efecte, una velocitat de 8.04Mbits/sec , la meitat que a l'anterior escenari.

Tercer escenari

En aquest escenari es segueix exactament el mateix procediment que al segon escenari, però aquest cop es dividiran els recursos de forma diferent que al segon escenari. L'enllaç

de pujada de la *slice* 2 gaudeix del 75% dels recursos i l'enllaç de baixada de la *slice* 2 gaudeix del 66% dels recursos. Amb aquest experiment es demostra que no falta que les *slices* de pujada o de baixada hagin de tenir el mateix percentatge de l'espectre per a funcionar correctament.

Per a fer aquest canvi només s'hi ha de modificar l'arxiu *ran-sharing.json* i canviar el nombre de *Resource Blocks* assignats a cada *slice* com s'ha vist al segon escenari. Aquest canvi afavoreix a la *slice* amb identificador 2 i, per tant, l'assignació de l'UE es realitza a aquesta *slice*. L'arxiu *ran-sharing.json* queda de la següent forma per l'*uplink*:

```
"id": 0,
  "label": "default",
  "static": {
    "posLow": 1,
    "posHigh": 5
  }
"id": 2,
  "label": "two",
  "static": {
    "posLow": 6,
    "posHigh": 25
  }
}
```

I, pel *downlink*:

```
"id": 0,
  "label": "default",
  "static": {
    "posLow": 0,
    "posHigh": 3
  }
"id": 2,
  "label": "default",
  "static": {
    "posLow": 4,
    "posHigh": 12
  }
}
```

I, l'arxiu *ue-association.json*, queda com:

```
"ueConfig": [
  {
    "imsi": 208920100001102,
    "dlSliceId": 2,
    "ulSliceId": 2
  }
]
```

Un cop modificats els arxius, els resultats han sigut els següents:

```

dlego@epc:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 172.16.0.2, port 32798
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 32800
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-1.00    sec 379 KBytes  3.10 Mbits/sec
[ 5] 1.00-2.00    sec 495 KBytes  4.05 Mbits/sec
[ 5] 2.00-3.00    sec 689 KBytes  5.64 Mbits/sec
[ 5] 3.00-4.00    sec 816 KBytes  6.68 Mbits/sec
[ 5] 4.00-5.00    sec 816 KBytes  6.68 Mbits/sec
[ 5] 5.00-6.00    sec 816 KBytes  6.68 Mbits/sec
[ 5] 6.00-7.00    sec 816 KBytes  6.68 Mbits/sec
[ 5] 7.00-8.00    sec 816 KBytes  6.68 Mbits/sec
[ 5] 8.00-9.00    sec 816 KBytes  6.68 Mbits/sec
[ 5] 9.00-10.00   sec 816 KBytes  6.68 Mbits/sec
[ 5] 10.00-10.62  sec 509 KBytes  6.68 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5] 0.00-10.62   sec 9.58 MBytes 7.57 Mbits/sec 0
[ 5] 0.00-10.62   sec 7.60 MBytes 6.00 Mbits/sec
-----
sender
receiver

```

Figura 5.14.: *Uplink* al 75%.

```

dlego@epc:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 172.16.0.2, port 36512
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 36514
[ ID] Interval      Transfer    Bandwidth    Retr  Cwnd
[ 5] 0.00-1.00    sec 465 KBytes  3.81 Mbits/sec 0 35.4 KBytes
[ 5] 1.00-2.00    sec 614 KBytes  5.03 Mbits/sec 0 63.6 KBytes
[ 5] 2.00-3.00    sec 1001 KBytes 8.20 Mbits/sec 0 112 KBytes
[ 5] 3.00-4.00    sec 1.27 MBytes 10.7 Mbits/sec 0 168 KBytes
[ 5] 4.00-5.00    sec 1.39 MBytes 11.7 Mbits/sec 0 233 KBytes
[ 5] 5.00-6.00    sec 1.47 MBytes 12.3 Mbits/sec 0 301 KBytes
[ 5] 6.00-7.00    sec 1.25 MBytes 10.5 Mbits/sec 0 365 KBytes
[ 5] 7.00-8.00    sec 1.43 MBytes 12.0 Mbits/sec 0 427 KBytes
[ 5] 8.00-9.00    sec 1.18 MBytes 9.87 Mbits/sec 0 494 KBytes
[ 5] 9.00-10.00   sec 1.50 MBytes 12.6 Mbits/sec 0 556 KBytes
[ 5] 10.00-10.06  sec 0.00 Bytes 0.00 bits/sec 0 556 KBytes
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5] 0.00-10.06   sec 11.5 MBytes 9.61 Mbits/sec 0
[ 5] 0.00-10.06   sec 10.6 MBytes 8.88 Mbits/sec
-----
sender
receiver

```

Figura 5.15.: *Downlink* al 66%.

Tant per l' *Uplink* (6.00Mbits/sec) com pel *Downlink* (9.61Mbits/sec) s'experimenta un augment de la velocitat en comparació amb l'anterior escenari.

Quart escenari

I, finalment, a l'últim escenari s'avalua el rendiment de la xarxa quan l'UE només gaudeix del 25% dels recursos de la xarxa per l' *Uplink* i del 33% dels recursos pel *Downlink*. En aquest cas l'arxiu *ran-sharing.json* no es modifica respecte al tercer escenari, però l'arxiu *ue-association.json* es canvia per tal d'assignar l'UE a la *slice* 0.

Per tant, l'arxiu *ue-association.json*, queda com:

```
"ueConfig": [
  {
    "imsi": 208920100001102,
    "dlSliceId": 0,
    "ulSliceId": 0
  }
]
```

Els resultats són els següents:

```
diego@epc:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 172.16.0.2, port 33232
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 33234
[ ID] Interval      Transfer    Bandwidth
[ 5]  0.00-1.00    sec 252 KBytes  2.06 Mbits/sec
[ 5]  1.00-2.00    sec 286 KBytes  2.34 Mbits/sec
[ 5]  2.00-3.00    sec 287 KBytes  2.35 Mbits/sec
[ 5]  3.00-4.00    sec 287 KBytes  2.35 Mbits/sec
[ 5]  4.00-5.00    sec 286 KBytes  2.34 Mbits/sec
[ 5]  5.00-6.00    sec 287 KBytes  2.35 Mbits/sec
[ 5]  6.00-7.00    sec 287 KBytes  2.35 Mbits/sec
[ 5]  7.00-8.00    sec 286 KBytes  2.34 Mbits/sec
[ 5]  8.00-9.00    sec 286 KBytes  2.34 Mbits/sec
[ 5]  9.00-10.00   sec 287 KBytes  2.35 Mbits/sec
[ 5] 10.00-11.00   sec 287 KBytes  2.35 Mbits/sec
[ 5] 11.00-11.53   sec 151 KBytes  2.35 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5]  0.00-11.53   sec 4.68 MBytes  3.41 Mbits/sec    0
[ 5]  0.00-11.53   sec 3.19 MBytes  2.32 Mbits/sec    0
-----
sender
receiver
```

Figura 5.16.: *Uplink* al 25%.

```
diego@epc:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 172.16.0.2, port 36296
[ 5] local 192.168.1.160 port 5201 connected to 172.16.0.2 port 36298
[ ID] Interval      Transfer    Bandwidth    Retr  Cwnd
[ 5]  0.00-1.00    sec 395 KBytes  3.23 Mbits/sec    0   31.1 KBytes
[ 5]  1.00-2.00    sec 595 KBytes  4.88 Mbits/sec    0   58.0 KBytes
[ 5]  2.00-3.00    sec 706 KBytes  5.78 Mbits/sec    0   90.5 KBytes
[ 5]  3.00-4.00    sec 697 KBytes  5.71 Mbits/sec    0   124 KBytes
[ 5]  4.00-5.00    sec 757 KBytes  6.20 Mbits/sec    0   160 KBytes
[ 5]  5.00-6.00    sec 793 KBytes  6.50 Mbits/sec    0   195 KBytes
[ 5]  6.00-7.00    sec 650 KBytes  5.33 Mbits/sec    0   228 KBytes
[ 5]  7.00-8.00    sec 799 KBytes  6.54 Mbits/sec    0   274 KBytes
[ 5]  8.00-9.00    sec 901 KBytes  7.38 Mbits/sec    0   372 KBytes
[ 5]  9.00-10.00   sec 1003 KBytes 8.21 Mbits/sec    0   499 KBytes
[ 5] 10.00-10.05   sec 0.00 Bytes  0.00 bits/sec    0   503 KBytes
-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 5]  0.00-10.05   sec 7.12 MBytes  5.95 Mbits/sec    0
[ 5]  0.00-10.05   sec 6.25 MBytes  5.22 Mbits/sec    0
-----
sender
receiver
```

Figura 5.17.: *Downlink* al 33%.

Tant per l'*uplink* (3.41Mbits/sec) com pel *downlink* (5.95Mbits/sec) s'experimenta una disminució de la velocitat en comparació amb l'anterior escenari.

6. Conclusions

En aquest Treball de Final de Grau s'ha dut a terme un prototip de xarxa LTE utilitzant el programari de codi obert OAI i Mosaic5G i la base de dades d'usuaris proporcionada per *Open Cells*. Amb aquesta implementació s'han complit tots els objectius marcats al projecte. S'ha aconseguit realitzar la configuració inicial de les dues màquines encarregades de fer les funcions d'eNB i d'EPC, muntar un escenari funcional el més fidelment possible als estàndards del 3GPP i, finalment, s'ha pogut monitorar la xarxa i segmentar els seus recursos tal com succeeix a la realitat.

En primer lloc, s'ha volgut realitzar una configuració inicial a l'estació de treball encarregada de dur a terme totes les funcions d'una xarxa LTE. Aquest pas previ a la realització del prototip s'ha pogut dur a terme gràcies a les especificacions que OAI proporciona. Amb aquest procediment el projecte ha començat a tomar forma i s'ha pogut observar la quantitat de recursos que són necessaris perquè una xarxa LTE, encara que sigui petita, pugui realitzar totes les seves funcions.

En segon lloc, amb la configuració inicial terminada, s'ha volgut realitzar la configuració dels dos elements principal de la xarxa: l'eNB i l'EPC. Amb aquest procediment s'han assentat els coneixements sobre les xarxes mòbils 4G a més dels coneixements obtinguts de la instal·lació, configuració i funcionament del *software* d'OAI obtinguts de Mosaic5G. Gràcies a aquest pas s'ha aconseguit realitzar un prototip de xarxa funcional a l'espera de la configuració de l'UE.

Amb els dos elements principals de la xarxa configurats i sent capaços de comunicar-se entre si s'ha volgut realitzar la configuració d'un terminal amb la intenció que utilitzi els recursos que proporciona el prototip. Aquest procediment s'ha pogut dur a terme gràcies als tutorials que proporciona OAI i la base de dades que *Open Cells* dona al públic. Amb aquest pas s'ha pogut connectar l'UE a internet emprant el prototip i, d'aquesta forma, verificar que s'ha dut a terme una xarxa LTE funcional preparada pel seu estudi.

En tercer lloc, s'ha volgut realitzar un estudi de la xarxa per tal de verificar la connexió entre els diferents elements de la xarxa i comprovar que realment l'UE es connecta a internet mitjançant el prototip de xarxa LTE. Aquest procediment s'ha dut a terme mitjançant l'analitzador de protocols *wireshark*. Amb aquest pas s'ha pogut comprovar tant els missatges de control i senyalització intercanviats entre l'eNB i l'EPC com els missatges al pla d'usuari que crea l'UE quan es connecta a internet i accedeix a determinants serveis.

En quart lloc, s'ha volgut realitzar un monitoratge en temps real emprant l'eina de monitoratge que proporciona Mosaic5G: *FlexRAN*. Amb aquest procediment s'ha pogut entendre com funciona l'eina i aprendre d'una forma més profunda com funciona el prototip de xarxa desenvolupat. A més de poder verificar tots els canvis que es produeixen a una xarxa LTE.

En cinquè lloc, s'ha volgut dividir l'espectre en diferents *slices* utilitzant l'eina *FlexRAN*. Amb aquest procediment s'ha pogut separar els recursos que pot proporcionar el prototip i assignar l'UE a una *slice*. Aquest pas s'ha pogut observar que no sempre cal proporcionar tots els recursos possibles a l'usuari per tal de donar un bon servei sinó que s'han d'assignar depenent dels que necessiti l'usuari per a poder dur a terme la petició que ha demanat.

7. Treball futur

Encara que s'han complit tots els objectius proposats al projecte, el prototip de xarxa es pot complementar i millorar de les següents formes:

- Implementació en dues màquines diferents l'eNB i de l'EPC per tal de millorar el rendiment dels dos elements i així alliberar la càrrega de treball de l'ordinador empleat al projecte.
- Configuració d'una segona estació base per tal de crear una xarxa d'eNBs emprant la interfície de xarxa X2 que no ha estat usada en aquest projecte.
- Augmentar el nombre d'UEs que es connecten a la xarxa per tal de comprovar el rendiment del prototip quan hi ha de proporcionar servei a més d'un usuari i, d'aquesta forma, fer un estudi més fidel de la creació i assignació de *slices* dependent dels recursos que necessiti cada dispositiu.
- Millorar la cobertura màxima que pot donar el prototip amb l'adquisició i configuració d'antenes. Aquestes juntament amb l'USR-P-B205mini empleat al projecte estarien encarregades de la transmissió i recepció de dades entre la xarxa i l'UE.

Bibliografía

- [1] Disable hyperthreading from within Linux (no access to BIOS). (2011, 15 febrero). Server Fault. <https://serverfault.com/questions/235825/disable-hyperthreading-from-within-linux-no-access-to-bios>
- [2] Ettus Research, a National Instruments Brand. (s. f.). Ettus Research - The leader in Software Defined Radio (SDR). Ettus Research. <https://www.ettus.com/>
- [3] M, J. (2016, 4 abril). How to Install Grub Customizer 5.0.5 in Ubuntu 16.04 LTS. UbuntuHandbook. <https://ubuntuhandbook.org/index.php/2016/04/install-grub-customizer-ubuntu-16-04-lts/>
- [4] P. (2019, 3 junio). Cómo cambiar el nombre del equipo en Linux desde el terminal. Linux Adictos. <https://www.linuxadictos.com/como-cambiar-el-nombre-del-equipo-en-linux-desde-el-terminal.html>
- [5] U. (2016, 22 agosto). How To: Install/Upgrade to Linux Kernel 4.7.1 in Ubuntu/Linux Mint Systems. YourOwnLinux. <http://www.yourownlinux.com/2016/08/how-to-install-linux-kernel-4-7-1-in-linux.html>
- [6] bibing. (2012). La evolución de la telefonía móvil. <http://bibing.us.es/proyectos/abreproy/11980/fichero/CAP>
- [7] Statista. (2020, 2 septiembre). Número de líneas de telefonía móvil pospago España 2007-2019. <https://es.statista.com/estadisticas/554075/numero-de-lineas-de-telefonía-movil-pospago-en-espana/>
- [8] Statista. (2020a, julio 9). LTE mobile subscriptions worldwide 2011-2025, by region. <https://www.statista.com/statistics/521572/4g-5g-mobile-subscriptions-worldwide/>
- [9] Expertos, E. (2018, 10 octubre). ¿Qué es GSM y cómo funciona? | VIU. Universidad Intenacional de Valencia. <https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-gsm-y-como-funciona>
- [10] GSM ALLIANCE. (2018). GSMA. <https://www.gsma.com/>
- [11] Mobile Economy. (2018). GSMA Mobile Economy. <https://www.gsma.com/mobileeconomy/>
- [12] S. (2015, 29 abril). Significado de TDMA. Significados. <https://www.significados.com/tdma/>

- [13] ¿Qué es una tarjeta SIM? (2019). Mas adelante. <https://www.masadelante.com/faqs/tarjeta-sim>
- [14] BBC News Mundo. (2018, 22 enero). Qué es el código IMEI y cómo usarlo para bloquear y desbloquear tu celular. <https://www.bbc.com/mundo/noticias-42774859>
- [15] EcuRed. (2018). WCDMA - EcuRed. <https://www.ecured.cu/WCDMA>
- [16] PCMag. (2018). Definition of 2G/3G architecture. <https://www.pcmag.com/encyclopedia/term/2g3g-architecture>
- [17] LTE Network Infrastructure and Elements - lteencyclopedia. (2018). LTE Encyclopedia. <https://sites.google.com/site/lteencyclopedia/lte-network-infrastructure-and-elements>
- [18] LTE - Tutorial 4G. (2014). IPV6GO. <http://www.ipv6go.net/lte/>
- [19] LTE Radio Protocol Architecture - Tutorialspoint. (2018). Tutorials Point. https://www.tutorialspoint.com/lte/lte_radio_protocol_architecture.htm
- [20] OpenAirInterface – 5G software alliance for democratising wireless innovation. (2014). Open Air. <https://www.openairinterface.org/>
- [21] I. (2019a, mayo 3). OpenAirInterface: An open platform for establishing the 5G system of the future. I'MTech. <https://imtech.wp.imt.fr/en/2017/05/23/openairinterface-5g-system-of-future/>
- [22] M. (2016a). Mosaic5G. Mosaic5G. <https://mosaic5g.io/>
- [23] M. (2016a). Flexran · Mosaic5G. FlexRAN. <https://mosaic5g.io/flexran/>
- [24] 4G and 5G reference software. (2017). OpenCells. <https://open-cells.com/>
- [25] About – 4G and 5G reference software. (2017). Open Cells. <https://open-cells.com/index.php/about/>
- [26] Laurent, A. (2019, 22 septiembre). All in one OpenAirInterface – 4G and 5G reference software. AllinOne. <https://open-cells.com/index.php/2019/09/22/all-in-one-openairinterface/>
- [27] B., G. (2020, 3 diciembre). ¿Qué es MySQL? Explicación detallada para principiantes. Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-mysql/>
- [28] OpenAirSystemRequirements · Wiki · oai / openairinterface5G. (2020, 15 junio). GitLab. <https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/OpenAirSystemRequirements>
- [29] OpenAirKernelMainSetup · Wiki · oai / openairinterface5G. (2019, 16 abril). GitLab. <https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/OpenAirKernelMainSetup>

-
- [30] ¿Qué es Hyper-Threading? (2018). Intel. <https://www.intel.es/content/www/es/es/gaming/resources/threading.html>
- [31] IBM Knowledge Center. (2018). plmn. https://www.ibm.com/support/knowledgecenter/es/SSSHRK_
- [32] Length of MNC? (2012, 15 octubre). Telit. <https://www.telit.com/technical-forum/length-of-mnc/>
- [33] Usage of OP/OPc and Transport Key. (2013). BlogSpot. <https://diameter-protocol.blogspot.com/2013/06/usage-of-opopc-and-transport-key.html>
- [34] Setting up OAI-CN + OAI-RAN + Flexran(2 PC) Setup. (2018). Hackmd. https://hackmd.io/@HktC6nD_T2iHdpqyA1Kqjw/S1ogmIFEHRun-OAI-and-FlexRAN
- [35] Mosaic5G. (2019, 18 marzo). Mosaic5G - Leveraging an Ecosystem of 5G services [Diapositivas]. Mosaic5G.io. <http://mosaic5g.io/resources/mosaic5g-oai-snaps-tutorial.pdf>
- [36] HowToConnectCOTSUEwithOAIeNBNew · Wiki · oai / openairinterface5G. (2020, 7 febrero). GitLab. <https://gitlab.eurecom.fr/oai/openairinterface5g/-/wikis/HowToConnectCOTSUEwithOAIeNBNew>
- [37] Slicing. (2020, 11 febrero). GitLab. <https://gitlab.eurecom.fr/mosaic5g/mosaic5g/-/wikis/tutorials/slicing>
- [38] Nohrborg, M. (s. f.). LTE. Long Term Evolution (LTE) -3GPP. <https://www.3gpp.org/technologies/keywords-acronyms/98-lte>
- [39] LTE-Tutorial 4G EPC. (2014). Arquitectura de EPC. IPV6GO. http://www.ipv6go.net/lte/arquitectura_epc.php
- [40] LTE - Tutorial 4G. (2014b). Protocolos E-UTRAN. IPV6GO. http://www.ipv6go.net/lte/eutran_protocolos.php

Part II.
Annexos

A. Configuració inicial de la màquina eNB

A.1. Instal·lació d'un *Kernel lowlatency*

OAI recomana la utilització d'un *kernel lowlatency* (o de baixa latència) per al sistema operatiu *Ubuntu* on s'executi les funcions d'un eNB. Segons la guia que dona OAI, per al sistema operatiu *Ubuntu 16.04* s'ha d'instal·lar la versió *4.4.0-21-lowlatency*. Els passos a seguir són els següents:

El primer pas consisteix a comprovar que el sistema està actualitzat. Per a dur-ho a terme es realitza la següent comanda:

```
sudo apt update
```

Una vegada que el sistema està actualitzat es passa a obtenir el *kernel* recomanat per OAI:

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

Amb aquesta comanda només es descarrega i s'instal·la el *kernel*, però no es selecciona com el sistema principal per a iniciar l'estació de treball, és a dir, encara que es tingui el *kernel* de *lowlatency* descarregat l'ordinador no s'inicia amb aquesta versió. Per a seleccionar-la s'ha d'editar l'arxiu de configuració de sistema *grub*. Per aquest motiu, el segon pas, consisteix a obtenir el programa anomenat *grub-customizer* amb el qual es pot seleccionar el *kernel* desitjat. La instal·lació del paquet segueix els següents passos:

Primer s'ha d'obtenir el repositori del programa:

```
sudo add-apt-repository ppa:danielrichter2007/grub-customizer
```

Un cop que ja es té el repositori, s'ha d'actualitzar la llista de paquets i instal·lar el programa:

```
sudo apt-get update  
sudo apt-get install grub-customizer
```

Quan ja s'ha obtingut el programa únicament s'ha d'obrir l'aplicació, anar a la pestanya de *General settings* (o configuracions general) i en l'opció de *default entry* (o entrada predefinida) seleccionar la versió de *kernel* amb la que es vol iniciar la màquina. En aquest cas s'ha seleccionat la versió *4.4.0-21-lowlatency*.

L'últim pas consisteix a reiniciar l'estació de treball. Quan l'ordinador estigui de nou encès, per a comprovar que la instal·lació del nou *kernel* ha estat correcta, s'executa la comanda:

```
uname -r
```

I la resposta que ens ha de donar la terminal és:

```
4.4.0-21-lowlatency
```

A.2. Control de la gestió de potència

OAI també recomana deshabilitar totes aquelles funcions de gestió d'energia del sistema operatiu (com els *C-States* i els *P-States*), l'escalat de freqüència de la CPU *Intel SpeedStep* i l'*Hyper-Threading*. Aquestes accions es poden fer tant des de la BIOS del sistema com des del fitxer de configuració *grub*. En aquest projecte s'ha seleccionat la segona opció.

El primer pas consisteix a obrir l'arxiu de configuració de sistema *grub*:

```
sudo emacs /etc/default/grub
```

Un cop que està obert es passa a editar-ho. Per tal de desactivar totes les funcions de gestió de l'energia s'ha d'editar el paràmetre *GRUB_CMDLINE_LINUX_DEFAULT* de la següent forma:

```
GRUB_CMDLINE_LINUX_DEFAULT="intel_pstate=disable processor.max_cstate=1  
intel_idle.max_cstate=0 idle=poll noht"
```

A continuació, s'han de guardar les modificacions, tancar l'editor de text *emacs* i actualitzar la configuració del sistema amb la comanda:

```
sudo update-grub
```

Un cop que es té els *C-States* i els *P-States* desactivats es passa a desactivar l'*Hyper-Threading*. Per a fer-ho es crea un arxiu de zero, el qual s'anomena *hyper*. Aquest arxiu és un programa escrit en *bash* amb permisos d'execució que s'encarrega de deshabilitar l'*Hyper-Threading*. El contingut de l'arxiu és el següent:

```
#!/bin/bash
for CPU in /sys/devices/system/cpu/cpu[0-9]*; do
    CPUID='basename $CPU | cut -b4-'
    echo -en "CPU: $CPUID\t"
    [ -e $CPU/online ] && echo "1" > $CPU/online
    THREAD1='cat $CPU/topology/thread_siblings_list | cut -f1 -d,'
    if [ $CPUID = $THREAD1 ]; then
        echo "-> enable"
        [ -e $CPU/online ] && echo "1" > $CPU/online
    else
        echo "-> disable"
        echo "0" > $CPU/online
    fi
done
```

El programa no és d'elaboració pròpia i s'ha obtingut de la pàgina web *serverfault*. La pàgina es pot consultar a la primera referència.

Un cop ja s'ha creat l'arxiu, se li han de donar permisos d'execució per tal que pugui realitzar la seva funció. La comanda per a dur-ho a terme és la següent:

```
chmod +x hyper
```

Finalment, per a executar-lo, s'ha de realitzar la següent comanda:

```
sudo ./hyper
```

El resultat que surt pel terminal és el següent:

```
CPU: 0 -> enable
CPU: 1 -> enable
CPU: 2 -> enable
CPU: 3 -> enable
CPU: 4 -> disable
CPU: 5 -> disable
CPU: 6 -> disable
CPU: 7 -> disable
```

D'aquesta forma s'ha desactivat *Hyper-Threading* de cada una de les 4 CPU que té l'estació de treball que s'usa al projecte. Cal destacar que cada cop que s'encengui l'estació de treball s'ha d'executar el programa.

El següent pas consisteix a desactivar l'escalat de freqüències del processador. Per a dur-ho a terme, s'ha d'instal·lar el programa *cpufrequtils*. La comanda és la següent:

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

Un cop que el programa està instal·lat s'ha d'editar el seu fitxer de configuració:

```
sudo emacs /etc/default/cpufrequtils
```

I afegir el parametre:

```
GOVERNOR="preformance"
```

A continuació, es guarden els canvis i es tanca l'editor de text. Finalment, s'ha de desactivar el *demon* (o dimoni) *ondemand* del sistema per tal d'evitar que els canvis realitzats no s'eliminin quan es reinicï el sistema. Les comandes són:

```
sudo update-rc.d ondemand disable  
sudo /etc/init.d/cpufrequtils restart
```

En aquest punt és necessari reiniciar el sistema per a finalitzar l'aplicació d'alguns canvis que encara no s'hagin dut a terme.

Una vegada que s'hagi reiniciat el sistema s'ha d'instal·lar el programa *i7z* que mostra l'estat del processador a temps d'execució del sistema. D'aquesta forma es comprova si tots els canvis han estat correctes. La comanda per obtenir el programa és:

```
sudo apt install i7z
```

Per obrir l'aplicació s'executa la comanda:

```
sudo i7z
```

El programa mostra per la mateixa terminal l'estat del processador. Com que tots els nuclis estan treballant al 100% de la seva freqüència i que tots els estats *C* estan al 0% menys el *C0* que està al 100%, es pot determinar que els canvis han sigut correctes. El resultat es pot comprovar a la figura A.1:

```

Cpu speed from cpufreq 3292.00MHz
cpufreq might be wrong if cpufreq is enabled. To guess correctly try estimating via tsc
Linux's inbuilt cpu_khz code emulated now
True Frequency (without accounting Turbo) 3292 MHz
CPU Multiplier 33x || Bus clock frequency (BCLK) 99.76 MHz

Socket [0] - [physical cores=4, logical cores=4, max online cores ever=4]
TURBO DISABLED on 4 Cores, Hyper Threading OFF
Max Frequency without considering Turbo 3292.00 MHz (99.76 x [33])
Max TURBO Multiplier (if Enabled) with 1/2/3/4 Cores is 38x/38x/37x/37x
Real Current Frequency 3292.00 MHz [99.76 x 33.00] (Max of below)

```

Core [core-id]	Actual Freq (Mult.)	C0%	Halt(C1)%	C3 %	C6 %	Temp	VCore
Core 1 [0]:	3292.00 (33.00x)	100	0	0	0	82	1.0752
Core 2 [1]:	3292.00 (33.00x)	100	0	0	0	84	1.0745
Core 3 [2]:	3292.00 (33.00x)	100	0	0	0	82	1.0763
Core 4 [3]:	3292.00 (33.00x)	100	0	0	0	84	1.0758

```

C0 = Processor running without halting
C1 = Processor running with halts (States >C0 are power saver modes with cores idling)
C3 = Cores running with PLL turned off and core cache turned off
C6, C7 = Everything in C3 + core state saved to last level cache, C7 is deeper than C6
Above values in table are in percentage over the last 1 sec
[core-id] refers to core-id number in /proc/cpufreq
'Garbage Values' message printed when garbage values are read
Ctrl+C to exit

```

Figura A.1.: Resultat de l'aplicació *i7z*.

A.3. Configuració del *hostname*

En aquest cas, canviar el *hostname* no és estrictament necessari, però anomenant la màquina amb l'element de la xarxa LTE que ha de simular facilita el treball.

Per a canviar el *hostname* únicament s'ha d'usar la següent comanda per terminal:

```
sudo nmcli general hostname eNB
```

Per a comprovar que s'ha canviat correctament s'ha de reiniciar l'ordinador i executar la comanda per terminar:

```
hostname
```

I, el resultat, ha de ser:

```
eNB
```

A.4. Configuració d'una IP estàtica

Per tal de configurar una IP estàtica s'ha d'editar el fitxer */etc/network/interfaces*. La comanda per a fer-ho és:

```
sudo emacs /etc/network/interfaces
```

Un cop obert s'edita el fitxer de la següent forma:

```
auto enp3s0
iface enp3s0 inet static
    address 192.168.1.150
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameserver 192.168.1.1
    dns-nameserver 1.1.1.1
```

enp3s0 és la interfície de xarxa que s'utilitza al projecte i es configura amb la IP estàtica 192.168.1.150.

A continuació es guarden els canvis, es tanca el fitxer i es reinicia la xarxa amb la comanda:

```
systemctl restart network
```

Per a comprovar que s'han realitzat correctament els canvis s'executa per terminal la comanda:

```
ifconfig
```

I el resultat ha de ser el següent:

```
diego@eNB:~$ ifconfig
enp3s0  Link encap:Ethernet  direcciónHW fc:aa:14:ff:65:aa
        Direc. inet:192.168.1.150  Difus.:192.168.1.255  Másc:255.255.255.0
        Dirección inet6: fe80::feaa:14ff:feff:65aa/64  Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1600  Métrica:1
        Paquetes RX:230649  errores:0  perdidos:0  overruns:0  frame:0
        Paquetes TX:15108  errores:0  perdidos:0  overruns:0  carrier:0
        colisiones:0  long.colaTX:1000
        Bytes RX:32607513 (32.6 MB)  TX bytes:1661515 (1.6 MB)
```

Figura A.2.: Configuració de la IP estàtica a l'eNB.

B. Configuració inicial de la màquina EPC

B.1. Instal·lació de *VMWare* i creació de la màquina virtual

Al projecte s'usa una única màquina per a la realització del prototip de xarxa. Com que l'eNB i l'EPC necessiten dos *kernels* diferents s'instal·la una màquina virtual.

El primer pas per a crear la màquina virtual consisteix a descarregar la imatge ISO de *Ubuntu 16*. Un cop obtinguda es passa a la instal·lació de *VMWare* a *Ubuntu*. Per a fer-ho s'ha d'anar a la pàgina web oficial de *VMWare* i instal·lar la versió *VMWare Player*. El que s'obté és el fitxer: *VMwarePlayer16.1.017198959.x86_64.bundle* que és l'encarregat de fer la instal·lació de *VMWare*.

A aquest arxiu se li ha de donar permisos d'execució perquè pugui dur a terme la seva funció. La comanda és:

```
chmod +x VMware-Player-16.1.0-17198959.x86_64.bundle
```

I s'executa:

```
sudo ./VMware\-Player\-16.1.0\-17198959.x86_64.bundle
```

Un cop que s'ha instal·lat *VMWare* s'ha de crear la màquina virtual. Per tant, el primer que s'ha de fer és obrir l'aplicació *VMWare*. Un cop oberta només s'ha de donar a l'opció de crear una nova màquina virtual i seleccionar la imatge ISO d'*Ubuntu* que s'ha descarregat anteriorment.

A continuació s'ha de triar les especificacions que ha de tenir la màquina virtual. Aquestes especificacions són les ja nomenades a la taula de l'apartat 4.4.2.

Una vegada que ja es té la màquina virtual instal·lada es pot començar a realitzar la configuració inicial de la màquina.

B.2. Instal·lació d'un *kernel* amb suport de GTP

OAI requereix la instal·lació de la versió 4.7.0 o superior del *kernel* de Linux a la màquina encarregada de realitzar les funcions de l'EPC. El *kernel* de *linux* inclou el mòdul GTP de tunelització de dades al pla d'usuari a partir de la versió 4.7.0. Per tant, s'ha d'instal·lar qualsevol versió superior a aquesta.

El primer pas consisteix a descarregar el *kernel*. Les comandes són:

```
wget kernel.ubuntu.com/~kernel-ppa/mainline/v4.7.1/linux-headers-4.7.1-040701_4.7.1-040701
.201608160432_all.deb
wget kernel.ubuntu.com/~kernel-ppa/mainline/v4.7.1/linux-headers-4.7.1-040701-generic_4.7.1
-040701.201608160432_amd64.deb
wget kernel.ubuntu.com/~kernel-ppa/mainline/v4.7.1/linux-image-4.7.1-040701-generic_4.7.1
-040701.201608160432_amd64.deb
```

I, a continuació, instal·lar-ho:

```
sudo dpkg -i linux-headers-4.7.1*.deb linux-image-4.7.1*.deb
```

De la mateixa forma que ha passat amb l'eNB en aquest punt s'ha descarregat i instal·lat el *kernel*, però no s'ha seleccionat perquè s'iniciï el sistema amb aquesta versió. Per tant, també s'ha d'instal·lar l'aplicació *grub-customizer*. El procediment a partir d'aquest punt és totalment anàleg que a l'ANNEX A.1.

Un cop que s'ha repetit el mateix procés que a l'eNB es comprova el correcte canvi de *kernel* amb la instrucció:

```
uname -r
```

I la resposta que ha de donar la terminal ha de ser:

```
4.7.1-040701-generic
```

B.3. Configuració del *hostname*

A diferència de l'eNB, en aquesta màquina, sí que és important canviar el *hostname* de la màquina. El motiu és pel FQDN que empenen els diferents mòduls de *software* de l'EPC per a comunicar-se entre ells.

El primer pas consisteix a canviar el nom de la màquina igual que a l'eNB, usant la instrucció:

```
sudo nmcli general hostname epc
```


A continuació, s'ha d'editar el fitxer `/etc/hosts`. Aquest fitxer permet establir una relació entre noms de màquina i direccions IP. Per tant, en aquest fitxer s'ha d'establir el FQDN que s'utilitza al projecte amb la direcció IP on, els diferents mòduls, s'escolten. Primer s'ha d'obrir el fitxer:

```
sudo emacs /etc/hosts
```

Una vegada obert s'edita de la següent forma:

```
127.0.0.1 localhost
127.0.1.1 epc.openair4G.eur epc
127.0.1.1 hss.openair4G.eur hss
```

Finalment, es guarda el fitxer i es tanca el fitxer. Per a comprovar que s'ha canviat correctament el `hostname` i que s'ha creat el FQDN primer s'ha de reiniciar la màquina i després utilitzar la comanda:

```
hostname -f
```

Pel terminal s'ha de veure:

```
epc.openair4G.eur
```

B.4. Configuració d'una IP estàtica

La configuració de la IP estàtica a l'EPC és exactament igual que a l'eNB amb la diferència de que en aquesta màquina la interfície de xarxa empleada és `ens33` i l'adreça IP és 192.168.1.160.

El fitxer de configuració `etc/network/interfaces` s'ha d'editar de la següent forma:

```
auto ens33
iface ens33 inet static
    address 192.168.1.160
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameserver 192.168.1.1
    dns-nameserver 1.1.1.1
```

A continuació es guarden els canvis, es tanca el fitxer i es reinicia la xarxa amb la comanda:

```
systemctl restart network
```

Per a comprovar que s'han realitzat correctament els canvis s'executa per terminal la comanda:

```
ifconfig
```

I el resultat ha de ser el següent:

```
diego@epc:~$ ifconfig
ens33  Link encap:Ethernet  HWaddr 00:0c:29:7b:21:e4
        inet addr:192.168.1.160  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:fe7b:21e4/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1600  Metric:1
        RX packets:41827 errors:0 dropped:1 overruns:0 frame:0
        TX packets:3253 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:11307311 (11.3 MB)  TX bytes:564445 (564.4 KB)
```

Figura B.1.: Configuració de la IP estàtica a l'EPC.

C. Arxius de configuració del projecte

C.1. Arxiu de configuració de l'eNB

```
Active_eNBs = ( "eNB-Eurecom-LTEBox");
# Asn1_verbosity, choice in: none, info, annoying
Asn1_verbosity = "none";

eNBs =
(
{
////////// Identification parameters:
eNB_ID      = 0xe00;

cell_type   = "CELL_MACRO_ENB";

eNB_name    = "eNB-Eurecom-LTEBox";

// Tracking area code, 0x0000 and 0xffffe are reserved values
tracking_area_code = 1;

plmn_list = ( { mcc = 208; mnc = 92; mnc_length = 2; } );
#plmn_list = ( { mcc = 214; mnc = 23; mnc_length = 2; } );

tr_s_preference      = "local_mac"

////////// Physical parameters:

component_carriers = (
{
node_function          = "3GPP_eNODEB";
node_timing            = "synch_to_ext_device";
node_synch_ref         = 0;
frame_type             = "FDD";
tdd_config             = 3;
tdd_config_s           = 0;
prefix_type            = "NORMAL";
eutra_band             = 7;
downlink_frequency     = 2685000000L;
uplink_frequency_offset = -120000000;
Nid_cell               = 0;
N_RB_DL                = 25;
Nid_cell_mbsfn         = 0;
nb_antenna_ports      = 1;
nb_antennas_tx         = 1;
nb_antennas_rx         = 1;
tx_gain                = 90;
rx_gain                = 125;
```

```

pbch_repetition                                = "FALSE";
prach_root                                    = 0;
prach_config_index                            = 0;
prach_high_speed                              = "DISABLE";
prach_zero_correlation                        = 1;
prach_freq_offset                            = 2;
pucch_delta_shift                            = 1;
pucch_nRB_CQI                                = 0;
pucch_nCS_AN                                 = 0;
pucch_n1_AN                                  = 0;
pdsch_referenceSignalPower                   = -27;
pdsch_p_b                                    = 0;
pusch_n_SB                                    = 1;
pusch_enable64QAM                            = "DISABLE";
pusch_hoppingMode                            = "interSubFrame";
pusch_hoppingOffset                          = 0;
pusch_groupHoppingEnabled                    = "ENABLE";
pusch_groupAssignment                         = 0;
pusch_sequenceHoppingEnabled                 = "DISABLE";
pusch_nDMRS1                                 = 1;
phich_duration                               = "NORMAL";
phich_resource                                = "ONESIXTH";
srs_enable                                    = "DISABLE";
/* srs_BandwidthConfig                        =;
srs_SubframeConfig                          =;
srs_ackNackST                               =;
srs_MaxUpPts                                =;*/

pusch_p0_Nominal                             = -96;
pusch_alpha                                  = "AL1";
pucch_p0_Nominal                             = -104;
msg3_deltaPreamble                          = 6;
pucch_deltaF_Format1                        = "deltaF2";
pucch_deltaF_Format1b                       = "deltaF3";
pucch_deltaF_Format2                        = "deltaF0";
pucch_deltaF_Format2a                       = "deltaF0";
pucch_deltaF_Format2b                       = "deltaF0";

rach_numberOfRA_Preambles                   = 64;
rach_preamblesGroupAConfig                  = "DISABLE";
/*
rach_sizeOfRA_PreamblesGroupA               = ;
rach_messageSizeGroupA                      = ;
rach_messagePowerOffsetGroupB               = ;
*/
rach_powerRampingStep                       = 4;
rach_preambleInitialReceivedTargetPower     = -108;
rach_preambleTransMax                       = 10;
rach_raResponseWindowSize                   = 10;
rach_macContentionResolutionTimer          = 48;
rach_maxHARQ_Msg3Tx                         = 4;

```

```

pcch_default_PagingCycle          = 128;
pcch_nB                            = "oneT";
bcch_modificationPeriodCoeff      = 2;
ue_TimersAndConstants_t300        = 1000;
ue_TimersAndConstants_t301        = 1000;
ue_TimersAndConstants_t310        = 1000;
ue_TimersAndConstants_t311        = 10000;
ue_TimersAndConstants_n310        = 20;
ue_TimersAndConstants_n311        = 1;
ue_TransmissionMode                = 1;

//Parameters for SIB18
rxPool_sc_CP_Len                  = "normal";
rxPool_sc_Period                  = "sf40";
rxPool_data_CP_Len                = "normal";
rxPool_ResourceConfig_prb_Num     = 20;
rxPool_ResourceConfig_prb_Start   = 5;
rxPool_ResourceConfig_prb_End     = 44;
rxPool_ResourceConfig_offsetIndicator_present = "prSmall";
rxPool_ResourceConfig_offsetIndicator_choice = 0;
rxPool_ResourceConfig_subframeBitmap_present = "prBs40";
rxPool_ResourceConfig_subframeBitmap_choice_bs_buf = "00000000000000000000";
rxPool_ResourceConfig_subframeBitmap_choice_bs_size = 5;
rxPool_ResourceConfig_subframeBitmap_choice_bs_bits_unused = 0;
/* rxPool_dataHoppingConfig_hoppingParameter = 0;
   rxPool_dataHoppingConfig_numSubbands = "ns1";
   rxPool_dataHoppingConfig_rbOffset = 0;
   rxPool_commTxResourceUC-ReqAllowed = "TRUE";
*/

// Parameters for SIB19
discRxPool_cp_Len                 = "normal"
discRxPool_discPeriod              = "rf32"
discRxPool_numRetx                 = 1;
discRxPool_numRepetition           = 2;
discRxPool_ResourceConfig_prb_Num  = 5;
discRxPool_ResourceConfig_prb_Start = 3;
discRxPool_ResourceConfig_prb_End  = 21;
discRxPool_ResourceConfig_offsetIndicator_present = "prSmall";
discRxPool_ResourceConfig_offsetIndicator_choice = 0;
discRxPool_ResourceConfig_subframeBitmap_present = "prBs40";
discRxPool_ResourceConfig_subframeBitmap_choice_bs_buf = "f0fffffff";
discRxPool_ResourceConfig_subframeBitmap_choice_bs_size = 5;
discRxPool_ResourceConfig_subframeBitmap_choice_bs_bits_unused = 0;

}
);
srbl_parameters :
{
    # timer_poll_retransmit = (ms) [5, 10, 15, 20,... 250, 300, 350, ... 500]
    timer_poll_retransmit = 80;
}

```

```

# timer_reordering = (ms) [0,5, ... 100, 110, 120, ... ,200]
timer_reordering      = 35;

# timer_reordering = (ms) [0,5, ... 250, 300, 350, ... ,500]
timer_status_prohibit = 0;

# poll_pdu = [4, 8, 16, 32 , 64, 128, 256, infinity(>10000)]
poll_pdu              = 4;

# poll_byte = (kB) [25,50,75,100,125,250,375,500,750,1000,1250,1500,2000,3000,
infinity(>10000)]
poll_byte             = 99999;

# max_retx_threshold = [1, 2, 3, 4 , 6, 8, 16, 32]
max_retx_threshold    = 4;
}

# ----- SCTP definitions
SCTP :
{
    # Number of streams to use in input/output
    SCTP_INSTREAMS = 2;
    SCTP_OUTSTREAMS = 2;
};

////////// MME parameters:
mme_ip_address      = ( { ipv4      = "192.168.1.160";
                        ipv6      = "192:168:30::17";
                        active    = "yes";
                        preference = "ipv4";
                        }
);

enable_measurement_reports = "no";
///X2
enable_x2 = "no";
t_reloc_prep      = 1000;      /* unit: millisecond */
tx2_reloc_overall = 2000;      /* unit: millisecond */

NETWORK_INTERFACES :
{
    ENB_INTERFACE_NAME_FOR_S1_MME      = "wlp2s0"; # "enp3s0";
    ENB_IPV4_ADDRESS_FOR_S1_MME        = "192.168.1.150/24";
    ENB_INTERFACE_NAME_FOR_S1U         = "wlp2s0"; # "enp3s0";
    ENB_IPV4_ADDRESS_FOR_S1U           = "192.168.1.150/24";
    ENB_PORT_FOR_S1U                   = 2152; # Spec 2152

    ENB_IPV4_ADDRESS_FOR_X2C            = "192.168.1.150/24";#"127.0.15.1/24";
    ENB_PORT_FOR_X2C                    = 36422; # Spec 36422
};

```

```
}
);
DU = (
{
DU_INTERFACE_NAME_FOR_F1U      = "lo";
DU_IPV4_ADDRESS_FOR_F1U       = "127.0.0.1/16";
DU_PORT_FOR_F1U               = 22100;
F1_U_DU_TRANSPORT_TYPE        = "TCP";
}
);
CU = (
{
CU_INTERFACE_NAME_FOR_F1U      = "lo";
CU_IPV4_ADDRESS_FOR_F1U       = "127.0.0.1"; //Address to search the DU
CU_PORT_FOR_F1U               = 22100;
F1_U_CU_TRANSPORT_TYPE        = "TCP";      // One of TCP/UDP/SCTP
DU_TYPE                        = "LTE";
}//,
// {
//     CU_INTERFACE_NAME_FOR_F1U      = "eth0";
//     CU_IPV4_ADDRESS_FOR_F1U       = "10.64.93.142"; //Address to search the DU
//     CU_PORT_FOR_F1U               = 2211;
//     F1_U_CU_TRANSPORT_TYPE        = "TCP";      // One of TCP/UDP/SCTP
//     DU_TYPE                        = "WiFi";
// }
);

CU_BALANCING = "ALL";
MACRLCs = (
{
num_cc = 1;
tr_s_preference = "local_L1";
tr_n_preference = "local_RRC";
phy_test_mode = 0;
    puSch10xSnr      = 200;
    puCch10xSnr     = 200;
}
);
L1s = (
{
num_cc = 1;
tr_n_preference = "local_mac";
}
);
RUs = (
{
    local_rf      = "yes"
    nb_tx        = 1
    nb_rx        = 1
}
```



```

    att_tx          = 0
    att_rx          = 0;
    bands           = [7];
    max_pdschReferenceSignalPower = -27;
    max_rxgain      = 125;
    eNB_instances   = [0];

}

);

NETWORK_CONTROLLER :
{
    FLEXRAN_ENABLED      = "yes";
    FLEXRAN_INTERFACE_NAME = "lo";
    FLEXRAN_IPV4_ADDRESS = "127.0.0.1";
    FLEXRAN_PORT         = 2210;
    FLEXRAN_CACHE        = "/mnt/oai_agent_cache";
    FLEXRAN_AWAIT_RECONF = "no";
};

# NETWORK_CONTROLLER :
# {
#     FLEXRAN_ENABLED      = "yes";
#     FLEXRAN_INTERFACE_NAME = "enp3s0";
#     FLEXRAN_INTERFACE_NAME = "wlp2s0";
#     FLEXRAN_IPV4_ADDRESS = "192.168.1.160";
#     FLEXRAN_PORT         = 2210;
#     FLEXRAN_CACHE        = "/mnt/oai_agent_cache";
#     FLEXRAN_AWAIT_RECONF = "no";
# };

THREAD_STRUCT = (
{
    #three config for level of parallelism "PARALLEL_SINGLE_THREAD", "PARALLEL_RU_L1_SPLIT",
    or "PARALLEL_RU_L1_TRX_SPLIT"
    parallel_config = "PARALLEL_RU_L1_TRX_SPLIT";
    #two option for worker "WORKER_DISABLE" or "WORKER_ENABLE"
    worker_config = "WORKER_ENABLE";
}
);

log_config :
{
    global_log_level      = "info";
    global_log_verbosity = "medium";
    hw_log_level          = "info";
    hw_log_verbosity      = "medium";
    phy_log_level         = "info";
    phy_log_verbosity     = "medium";
    mac_log_level         = "info";
};

```

```
mac_log_verbosity          = "high";
rlc_log_level              = "info";
rlc_log_verbosity          = "medium";
pdcp_log_level             = "info";
pdcp_log_verbosity        = "medium";
rrc_log_level              = "info";
rrc_log_verbosity         = "medium";
};
```


C.2. Arxius de configuració de *FlexRAN*

C.2.1. Creació de *slices*

```
{
  "ul": {
    "algorithm": "Static",
    "slices": [
      {
        "id": 0,
        "label": "default",
        "static": {
          "posLow": 1,
          "posHigh": 12
        }
      },
      {
        "id": 2,
        "label": "two",
        "static": {
          "posLow": 13,
          "posHigh": 23
        }
      }
    ]
  },
  "dl": {
    "algorithm": "Static",
    "slices": [
      {
        "id": 0,
        "label": "default",
        "static": {
          "posLow": 0,
          "posHigh": 5
        }
      },
      {
        "id": 2,
        "label": "two",
        "static": {
          "posLow": 6,
          "posHigh": 12
        }
      }
    ]
  }
}
```

C.2.2. Assignació de *slices*

```
{
  "ueConfig": [
    {
      "imsi": 208920100001102,
      "dlSliceId": 2,
      "ulSliceId": 2
    }
  ]
}
```

C.3. Arxius de configuració de l'EPC

C.3.1. Arxius de configuració de l'HSS

HSS

```
#####  
# Licensed to the OpenAirInterface (OAI) Software Alliance under one or more  
# contributor license agreements. See the NOTICE file distributed with  
# this work for additional information regarding copyright ownership.  
# The OpenAirInterface Software Alliance licenses this file to You under  
# the Apache License, Version 2.0 (the "License"); you may not use this file  
# except in compliance with the License.  
# You may obtain a copy of the License at  
#  
#   http://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#-----  
# For more information about the OpenAirInterface (OAI) Software Alliance:  
#   contact@openairinterface.org  
#####  
HSS :  
{  
  ## MySQL mandatory options  
  MYSQL_server = "127.0.0.1";      # HSS S6a bind address  
  MYSQL_user   = "root";          # Database server login  
  MYSQL_pass   = "linux";         # Database server password  
  MYSQL_db     = "oai_db";        # Your database name  
  ## HSS options  
  #OPERATOR_key = "1006020f0a478bf6b699f15c062e42b3"; # OP key matching your database  
  #OPERATOR_key = "11111111111111111111111111111111"; # OP key matching your database  
  OPERATOR_key = "";             # OP key matching your database  
  RANDOM = "true";              # True random or only pseudo random (for subscriber vector generation)  
  
  ## Freediameter options  
  FD_conf = "/var/snap/oai-cn/current/hss_fd.conf";  
};
```

HSS_fd

```

# ----- Local -----
# The first parameter in this section is Identity, which will be used to
# identify this peer in the Diameter network. The Diameter protocol mandates
# that the Identity used is a valid FQDN for the peer. This parameter can be
# omitted, in that case the framework will attempt to use system default value
# (as returned by hostname --fqdn).
Identity = "epc.openair4G.eur";
# In Diameter, all peers also belong to a Realm. If the realm is not specified,
# the framework uses the part of the Identity after the first dot.
Realm = "openair4G.eur";
# This parameter is mandatory, even if it is possible to disable TLS for peers
# connections. A valid certificate for this Diameter Identity is expected.
TLS_Cred = "/var/snap/oai-cn/current/hss.cert.pem", "/var/snap/oai-cn/current/hss.key.pem";
TLS_CA = "/var/snap/oai-cn/current/hss.cacert.pem";
# Disable use of TCP protocol (only listen and connect in SCTP)
# Default : TCP enabled
No_SCTP;
# This option is ignored if freeDiameter is compiled with DISABLE_SCTP option.
# Prefer TCP instead of SCTP for establishing new connections.
# This setting may be overwritten per peer in peer configuration blocs.
# Default : SCTP is attempted first.
Prefer_TCP;
# Disable use of IPv6 addresses (only IP)
# Default : IPv6 enabled
No_IPv6;

# Overwrite the number of SCTP streams. This value should be kept low,
# especially if you are using TLS over SCTP, because it consumes a lot of
# resources in that case. See tickets 19 and 27 for some additional details on
# this.
# Limit the number of SCTP streams
SCTP_streams = 3;
# By default, freeDiameter acts as a Diameter Relay Agent by forwarding all
# messages it cannot handle locally. This parameter disables this behavior.
NoRelay;
# Use RFC3588 method for TLS protection, where TLS is negotiated after CER/CEA exchange
# is completed
# on the unsecure connection. The alternative is RFC6733 mechanism, where TLS protects also
# the CER/CEA exchange on a dedicated secure port.
# This parameter only affects outgoing connections.
# The setting can be also defined per-peer (see Peers configuration section).
# Default: use RFC6733 method with separate port for TLS.

#TLS_old_method;

```

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

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

Port = 3868;
SecPort = 5868;

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

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

# Uncomment (and create acl.conf) to allow incoming connections from other peers.
LoadExtension = "acl_wl.fdx" : "/var/snap/oai-cn/current/acl.conf";

# Uncomment to display periodic state information
#LoadExtension = "dbg_monitor.fdx";
# Uncomment to enable an interactive Python interpreter session.
# (see doc/dbg_interactive.py.sample for more information)
#LoadExtension = "dbg_interactive.fdx";
# Load the RFC4005 dictionary objects
#LoadExtension = "dict_nasreq.fdx";

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

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

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

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

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

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

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


C.3.2. Arxius de configuració de l'MME

MME

```
#####
# Licensed to the OpenAirInterface (OAI) Software Alliance under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The OpenAirInterface Software Alliance licenses this file to You under
# the Apache License, Version 2.0 (the "License"); you may not use this file
# except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#-----
# For more information about the OpenAirInterface (OAI) Software Alliance:
#     contact@openairinterface.org
#####
MME :
{
    REALM                                = "openair4G.eur";    # YOUR REALM HERE
    PID_DIRECTORY                        = "/var/run";
    # Define the limits of the system in terms of served eNB and served UE.
    # When the limits will be reached, overload procedure will take place.
    MAXENB                               = 2;                    # power of 2
    MAXUE                                 = 16;                    # power of 2
    RELATIVE_CAPACITY                     = 10;

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

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

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

    IP_CAPABILITY = "IPv4v6"; # UNUSED, TODO
}
```

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

S6A :
{
# YOUR MME freeDiameter config file path
S6A_CONF                  = "/var/snap/oai-cn/current/mme_fd.conf";
# THE HSS HOSTNAME
HSS_HOSTNAME              = "epc";
};

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

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

# ----- MME served GUMMEIs
# MME code DEFAULT size = 8 bits
# MME GROUP ID size = 16 bits
GUMMEI_LIST = (
    {MCC="208" ; MNC="92"; MME_GID="4" ; MME_CODE="1"; }
    #{MCC="214" ; MNC="23"; MME_GID="4" ; MME_CODE="1"; }
# YOUR GUMMEI CONFIG HERE
);

# ----- MME served TAIs
# TA (mcc.mnc:tracking area code) DEFAULT = 208.34:1
# max values = 999.999:65535
# maximum of 16 TAIs, comma separated
# !!! Actually use only one PLMN
TAI_LIST = (
    {MCC="208" ; MNC="92"; TAC = "1"; }
    #{MCC="214" ; MNC="23"; MME_GID="4" ; MME_CODE="1"; }
# YOUR TAI CONFIG HERE
);
```

```

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

# EMM TIMERS
# T3402 start:
# At attach failure and the attempt counter is equal to 5.
# At tracking area updating failure and the attempt counter is equal to 5.
# T3402 stop:
# ATTACH REQUEST sent, TRACKING AREA REQUEST sent.
# On expiry:
# Initiation of the attach procedure, if still required or TAU procedure
# attached for emergency bearer services.
T3402                = 1 # in minutes (default is 12 minutes)

# T3412 start:
# In EMM-REGISTERED, when EMM-CONNECTED mode is left.
# T3412 stop:
# When entering state EMM-DEREGISTERED or when entering EMM-CONNECTED mode.
# On expiry:
# Initiation of the periodic TAU procedure if the UE is not attached for
# emergency bearer services. Implicit detach from network if the UE is
# attached for emergency bearer services.
T3412                = 54 # in minutes (default is 54 minutes,
network dependent)
# T3422 start: DETACH REQUEST sent
# T3422 stop: DETACH ACCEPT received
# ON THE 1st, 2nd, 3rd, 4th EXPIRY: Retransmission of DETACH REQUEST
T3422                = 6 # in seconds (default is 6s)

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

# T3460 start: AUTHENTICATION REQUEST sent, SECURITY MODE COMMAND sent
# T3460 stop:
# AUTHENTICATION RESPONSE received, AUTHENTICATION FAILURE received,
# SECURITY MODE COMPLETE received, SECURITY MODE REJECT received
# ON THE 1st, 2nd, 3rd, 4th EXPIRY: Retransmission of the same message type

```

```

T3460                                = 6 # in seconds (default is 6s)

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

# ESM TIMERS
T3485                                = 8 # UNUSED in seconds (default is 8s)
T3486                                = 8 # UNUSED in seconds (default is 8s)
T3489                                = 4 # UNUSED in seconds (default is 4s)
T3495                                = 8 # UNUSED in seconds (default is 8s)
};

NETWORK_INTERFACES :
{
# MME binded interface for S1-C or S1-MME communication (S1AP), can be ethernet
interface, virtual ethernet interface, we don't advise wireless interfaces
MME_INTERFACE_NAME_FOR_S1_MME        = "ens33"; # YOUR NETWORK CONFIG HERE
MME_IPV4_ADDRESS_FOR_S1_MME          = "192.168.1.160/24"; #YOUR NETWORK CONFIG HERE

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

LOGGING :
{
# OUTPUT choice in { "CONSOLE", "SYSLOG", 'path to file', "'IPv40':TCP port num'"
# 'path to file' must start with '.' or '/'
# if TCP stream choice, then you can easily dump the traffic on the remote or local
host: nc -l 'TCP port num' > received.txt
OUTPUT                                = "CONSOLE";
};

#OUTPUT                                = "SYSLOG";
#OUTPUT                                = "/tmp/mme.log";
#OUTPUT                                = "127.0.0.1:5656";

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

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

```

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

# ASN1 VERBOSITY: none, info, annoying
# for S1AP protocol
ASN1_VERBOSITY = "none";
};
TESTING :
{
# file should be copied here from source tree by following command:
run_mme --install-mme-files ...
SCENARIO_FILE = "/usr/local/share/oai/test/mme/no_regression.xml";
};
};

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


MME_fd

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

# Uncomment if the framework cannot resolv it.
Identity = "epc.openair4G.eur";
Realm = "openair4G.eur";

# TLS configuration (see previous section)
TLS_Cred = "/var/snap/oai-cn/current/mme.cert.pem",
          "/var/snap/oai-cn/current/mme.key.pem";
TLS_CA   = "/var/snap/oai-cn/current/mme.cacert.pem";

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

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

No_IPv6;

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

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

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

#TLS_old_method;

AppServThreads = 4;

# Specify the addresses on which to bind the listening server. This must be
# specified if the framework is unable to auto-detect these addresses, or if the
```

```
# auto-detected values are incorrect. Note that the list of addresses is sent
# in CER or CEA message, so one should pay attention to this parameter if some
# adresses should be kept hidden.
#ListenOn = ;
Port = 3870;
SecPort = 5870;
# ----- Extensions -----
# Uncomment (and create rtd.conf) to specify routing table for this peer.
#LoadExtension = "rt_default.fdx" : "rtd.conf";
# Uncomment (and create acl.conf) to allow incoming connections from other peers.
#LoadExtension = "acl_wl.fdx" : "acl.conf";
# Uncomment to display periodic state information
#LoadExtension = "dbg_monitor.fdx";
# Uncomment to enable an interactive Python interpreter session.
# (see doc/dbg_interactive.py.sample for more information)
#LoadExtension = "dbg_interactive.fdx";
# Load the RFC4005 dictionary objects
#LoadExtension = "dict_nasreq.fdx";
LoadExtension = "dict_nas_mipv6.fdx";
LoadExtension = "dict_s6a.fdx";
# Load RFC4072 dictionary objects
#LoadExtension = "dict_eap.fdx";
# Load the Diameter EAP server extension (requires diameap.conf)
#LoadExtension = "app_diameap.fdx" : "diameap.conf";
# Load the Accounting Server extension (requires app_acct.conf)
#LoadExtension = "app_acct.fdx" : "app_acct.conf";
# ----- Peers -----
# The framework will actively attempt to establish and maintain a connection
# with the peers listed here.
# For only accepting incoming connections, see the acl_wl.fdx extension.
# ConnectPeer
# Declare a remote peer to which this peer must maintain a connection.
# In addition, this allows specifying non-default parameters for this peer only
# (for example disable SCTP with this peer, or use RFC3588-flavour TLS).
# Note that by default, if a peer is not listed as a ConnectPeer entry, an
# incoming connection from this peer will be rejected. If you want to accept
# incoming connections from other peers, see the acl_wl.fdx? extension which
# allows exactly this.
ConnectPeer= "epc.openair4G.eur" { ConnectTo = "127.0.0.1";No_SCTP; No_IPv6;Prefer_TCP;No_TLS;
port = 3868; realm = "openair4G.eur"};
```


C.3.3. Arxiu de configuració de l'SPGW

```
#####
# Licensed to the OpenAirInterface (OAI) Software Alliance under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The OpenAirInterface Software Alliance licenses this file to You under
# the Apache License, Version 2.0 (the "License"); you may not use this file
# except in compliance with the License.
# You may obtain a copy of the License at
#   http://www.apache.org/licenses/LICENSE-2.0
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#-----
# For more information about the OpenAirInterface (OAI) Software Alliance:
#   contact@openairinterface.org
#####
S-GW :
{
  NETWORK_INTERFACES :
  {
    # S-GW binded interface for S11 communication (GTPV2-C), if none selected the ITTI
    message interface is used
    SGW_INTERFACE_NAME_FOR_S11           = "lo"; # STRING, interface name,
    YOUR NETWORK CONFIG HERE
    SGW_IPV4_ADDRESS_FOR_S11             = "127.0.11.2/8"; # STRING, CIDR,
    YOUR NETWORK CONFIG HERE
    # S-GW binded interface for S1-U communication (GTPV1-U) can be ethernet interface,
    virtual ethernet interface, we don't advise wireless interfaces
    SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP = "ens33"; # STRING, interface name,
    YOUR NETWORK CONFIG HERE, USE "lo" if S-GW run on eNB host
    SGW_IPV4_ADDRESS_FOR_S1U_S12_S4_UP   = "192.168.1.160/24"; # STRING, CIDR,
    YOUR NETWORK CONFIG HERE
    SGW_IPV4_PORT_FOR_S1U_S12_S4_UP      = 2152; # INTEGER, port
    number, PREFER NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE DOING
    # S-GW binded interface for S5 or S8 communication, not implemented, so leave it to
    none
    SGW_INTERFACE_NAME_FOR_S5_S8_UP      = "none"; # STRING, interface
    name, DO NOT CHANGE (NOT IMPLEMENTED YET)
    SGW_IPV4_ADDRESS_FOR_S5_S8_UP        = "0.0.0.0/24"; # STRING, CIDR, DO
    NOT CHANGE (NOT IMPLEMENTED YET)
    # external controller for managing context of UE and its user plane
    SGW_REMOTE_CONTROLLER_ENABLED = "no";
    SGW_REMOTE_CONTROLLER_IPV4_ADDRESS = "127.0.0.1";
    SGW_REMOTE_CONTROLLER_PORT = 9999;
  };
};
```

```

INTERTASK_INTERFACE :
{
    # max queue size per task
    ITTI_QUEUE_SIZE          = 2000000;                # INTEGER
};
LOGGING :
{
    # OUTPUT choice in { "CONSOLE", "SYSLOG", 'path to file', "'IPv4@':TCP port num'"
    # 'path to file' must start with '.' or '/'
    # if TCP stream choice, then you can easily dump the traffic on the remote or local
    host: nc -l 'TCP port num' > received.txt
    OUTPUT                   = "CONSOLE";            # see 3 lines above
    #OUTPUT                  = "SYSLOG";             # see 4 lines above
    #OUTPUT                  = "/tmp/spgw.log";      # see 5 lines above
    #OUTPUT                  = "127.0.0.1:5656";    # see 6 lines above

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

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

    # Log level choice in { "EMERGENCY", "ALERT", "CRITICAL", "ERROR", "WARNING", "NOTICE",
    "INFO", "DEBUG", "TRACE" }
    UDP_LOG_LEVEL            = "TRACE";
    GTPV1U_LOG_LEVEL        = "TRACE";
    GTPV2C_LOG_LEVEL        = "TRACE";
    SPGW_APP_LOG_LEVEL      = "TRACE";
    S11_LOG_LEVEL           = "TRACE";
};
};
P-GW =
{
    NETWORK_INTERFACES :
    {
        # P-GW binded interface for S5 or S8 communication, not implemented, so leave it to
        none
        PGW_INTERFACE_NAME_FOR_S5_S8          = "none"; # STRING, interface name, DO NOT
        CHANGE (NOT IMPLEMENTED YET)

        # P-GW binded interface for SGI (egress/ingress internet traffic)
        PGW_INTERFACE_NAME_FOR_SGI           = "ens33"; # STRING, YOUR NETWORK
        CONFIG HERE
        PGW_MASQUERADE_SGI                   = "yes"; # STRING, {"yes", "no"}. YOUR NETWORK
        CONFIG HERE, will do NAT for you if you put "yes".
        UE_TCP_MSS_CLAMPING                  = "no"; # STRING, {"yes", "no"}.
    };
};

```

```
# Pool of UE assigned IP addresses
# Do not make IP pools overlap
# first IPv4 address X.Y.Z.1 is reserved for GTP network device on SPGW
# Normally no more than 16 pools allowed, but since recent GTP kernel module use,
only one pool allowed (TODO).
IP_ADDRESS_POOL :
{
    IPV4_LIST = (
        "172.16.0.0/12" # STRING, CIDR, YOUR NETWORK CONFIG HERE.
    );
};

# DNS address communicated to UEs
DEFAULT_DNS_IPV4_ADDRESS = "1.1.1.1"; # "8.8.8.8"; # YOUR NETWORK CONFIG HERE
DEFAULT_DNS_SEC_IPV4_ADDRESS = "8.8.4.4"; # YOUR NETWORK CONFIG HERE

# Non standard feature, normally should be set to "no", but you may need to set to yes
for UE that do not explicitly request a PDN address through NAS signalling
FORCE_PUSH_PROTOCOL_CONFIGURATION_OPTIONS = "no"; # STRING, {"yes", "no"}.
UE_MTU = 1500 # INTEGER
};
```