



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa.

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Wilder Peña, Walter Piastri, Pablo Vázquez

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRICISTA.

TUTORES

Claudina Rattaro Universidad de la República
Lucas Inglés Universidad de la República

TRIBUNAL

Pablo Belzarena..... Universidad de la República
María Simon..... Universidad de la República
Álvaro Valdés..... Universidad de la República

Montevideo
20 de noviembre de 2023

Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa., Wilder Peña, Walter Piastri, Pablo Vázquez.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 196 páginas.
Compilada el sábado 10 febrero, 2024.
<http://iie.fing.edu.uy/>

Sean los orientales tan ilustrados como valientes.

JOSÉ GERVASIO ARTIGAS

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

Queremos expresar nuestra sincera gratitud al Ministerio de Educación y Cultura, a través del Fondo Vaz Ferreira (FVF-2021-128–DICYT), por su valioso apoyo financiero que hizo posible este proyecto. Agradecemos a ANTEL, por proporcionarnos un teléfono móvil que facilitó las pruebas con equipos comerciales. También a la Fundación Ricaldoni y a las autoridades de Facultad, por su apoyo y sus gestiones al momento de superar dificultades. A Claudina y Lucas, nuestros tutores, por acompañarnos en cada paso, aportando su visión crítica, y su apoyo permanente. A nuestras familias y su paciencia, por ser nuestro soporte y constante fuente de motivación en este camino. A todos los que contribuyeron, su respaldo fue fundamental para este logro.

Agradecemos también el interés y el reconocimiento del público y del tribunal hacia nuestro proyecto en la última edición de Ingeniería de Muestra en la que fuimos honrados con el primer premio del concurso de proyectos de grado del Instituto de Ingeniería Eléctrica.

Esta página ha sido intencionalmente dejada en blanco.

A Pierina, Ainoha y Alfonsina.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

Las redes móviles están en continua transformación para satisfacer las crecientes demandas de servicios diversos con requisitos distintos en cuanto a tráfico, calidad de servicio y eficiencia energética. Las redes de quinta generación (5G) son una realidad en pleno despliegue y alcanzando rápidamente una fuerte penetración y la madurez tecnológica. Las discusiones sobre las futuras redes 6G ya está en marcha, lo que subraya la importancia de la innovación en este campo.

Este desarrollo no se hace solo y no debe ser hecho solamente por otros. Participar en la investigación y en el debate técnico de estos temas hace a la esencia universitaria. Vincularlo a las necesidades de operadores, reguladores y desarrolladores nacionales es una condición necesaria para su evolución. Las investigaciones actuales en *scheduling*, aplicación de inteligencia artificial en la optimización de la asignación de recursos de radio, y de la integración de redes ópticas con redes móviles, van en el sentido de las necesidades de corto y mediano plazo.

Este proyecto busca aportar una herramienta de trabajo para las investigaciones actuales y futuras. Se implementa una maqueta de red 5G *Stand Alone* en código abierto, con interfaces de radio operativas. Se evalúa su funcionamiento, y se modifica cambiando el *scheduler* original, como ejercicio de validación de su potencialidad como entorno de desarrollo.

Se estudiaron los sistemas de código abierto existente, con desarrollos completos en 5G. Se evaluaron los sistemas de OpenAirInterface y srsRAN en forma documental y experimentando con ellos. También se revisaron las condiciones institucionales y de entorno de ambas, para buscar una solución con buena proyección de futuro. El trabajo se hizo en una etapa temprana, con pocos meses de lanzado el primero y durante el lanzamiento del segundo sistema estudiado.

Se instaló, estudió, configuró y evaluó el sistema elegido de OpenAirInterface. En el trabajo se detallan las características y opciones de configuración utilizadas, los escenarios con los que se probó y los resultados logrados. Las pruebas de *throughput* usando variaciones de configuración, mostraron resultados alentadores con tasas de transferencia de 70 Mb/s en configuraciones *Time Division Duplex* de una sola capa con 80 MHz de ancho de banda. También se conectaron teléfonos comerciales que accedieron a internet por la red. Se utilizaron servicios de videollamada entre dos teléfonos sin interrupciones ni inconvenientes.

El desarrollo exitoso de un *scheduler* alternativo simple permitió explorar la arquitectura del sistema y validarlo como la herramienta de desarrollo buscada. En conclusión, la continuidad del desarrollo de este trabajo y su uso como herramienta de investigación y laboratorio de enseñanza parecen pasos inmediatos inevitables.

Esta página ha sido intencionalmente dejada en blanco.

Prefacio

La evolución de la tecnología de red móvil ha transformado gradual y drásticamente la vida de todos. Este progreso se ha infiltrado en nuestro día a día y ha cambiado la forma en que nos comunicamos, aprovechamos el tiempo libre, aprendemos y trabajamos.

Más allá del avance tecnológico, el verdadero valor de la innovación radica en la búsqueda de un futuro mejor.

W.W.P.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos	III
Resumen	VII
Prefacio	IX
1. Introducción	1
1.1. Motivación y Antecedentes	1
1.2. Objetivo General	3
1.3. Objetivos Específicos	3
1.4. Alcance	4
1.5. Productos y Contribuciones del proyecto	5
1.6. Organización del Documento	7
2. Redes Móviles 5G	9
2.1. ¿Qué es 5G?	9
2.2. Estrategias de Despliegue	11
2.3. Características Principales	12
2.4. Conceptos Claves para la Implementación	17
2.4.1. Duplexado por División en Tiempo y Frecuencia	18
2.4.2. OFDM, Numerología y Estructura de Trama en 5G	20
2.4.3. Sincronización	25
2.4.4. El tamaño del Bloque de Transporte (TBS)	27
2.4.5. HARQ Retransmisiones Compartidas entre Capa Dos y Capa Física	29
2.5. Evolución y Desafíos	32
2.6. Resumen	36
3. Selección de Plataforma de Software	37
3.1. Las Plataformas Evaluadas	37
3.2. Factibilidad	39
3.3. Avance de los Desarrollos	39
3.3.1. srsRAN	39
3.3.2. OpenAirInterface	40
3.4. Instituciones, Foros y Actividades.	41
3.5. Instalaciones Preliminares de Prueba	42

Tabla de contenidos

3.6. Selección	43
4. Implementación	45
4.1. Core Network	45
4.2. Radio Access Network	46
4.3. Especificación del hardware	48
4.3.1. Servidores	48
4.3.2. Conexión con USRP	48
4.3.3. Interfaz de Radiofrecuencia	49
4.3.4. Backup de Energía	49
4.4. Opciones de Configuración	49
4.5. Escenarios de Uso	54
4.5.1. Sistemas con RFsimulator	54
4.5.2. Sistemas con SDR	56
4.6. Presupuesto	58
5. Evaluación de Performance	61
5.1. Pruebas de Conectividad	61
5.2. Pruebas de Throughput	63
5.3. Consideraciones Cualitativas Sobre el Hardware y la Estabilidad del Sistema	75
6. Desarrollo de un Scheduler Alternativo	77
6.1. Selección de la Aplicación a Desarrollar	77
6.1.1. Descripción de un <i>Scheduler</i>	77
6.2. Descripción de la Estructura de Software del Sistema	79
6.3. Algoritmo Original	82
6.4. Desarrollo	84
6.5. Cambio de Parámetros por Configuración	87
6.6. Ensayos y Resultados	87
6.7. Resumen	95
7. Conclusiones	97
7.1. Cumplimiento de Objetivos	97
7.2. Caracterización de la Maqueta	98
7.3. Los Próximos Pasos	98
A. Manuales	101
A.1. Datos del Hardware	101
A.1.1. USRP X310	101
A.1.2. Daughterboard UBX 160	102
A.1.3. LP0965 Antenna	103
A.1.4. Tarjeta Ethernet 10 Gbps X520-DA2	103
A.1.5. Servidores	105
A.2. Instalación de la plataforma	105
A.2.1. 5GCN	106

A.2.2. gNB	108
A.2.3. UE	109
A.2.4. Ejecutar la Plataforma Completa	109
A.2.5. RF Simulator	110
A.2.6. 2 UE con <i>Network Name Spaces</i>	112
A.3. Procedimientos	117
A.3.1. Instalar y Correr un Sistema Completo en Dockers	117
A.3.2. Archivo de Configuración Comentado	119
A.3.3. Cambiar Banda, Frecuencia o Ancho de Banda en un gNB	125
A.3.4. Procedimiento de Pruebas de SDR	128
A.4. Herramientas Adicionales	132
B. Instalación de Plataforma srsRAN	133
B.1. Instalación de radios virtuales	133
B.2. Instalación del core Open5gs	134
B.3. Ejecución	135
B.4. Pruebas de Funcionamiento srsRAN ZeroMQ Virtual Radio	135
B.4.1. Objetivo	135
B.4.2. Ejecución	135
B.4.3. Conclusión	137
C. Protocolo de Pruebas y Reporte de Resultados	139
C.1. Protocolo de Pruebas	139
C.2. Reporte de resultados	140
D. Scheduler Alternativo	149
D.1. Readme	149
D.2. Modificaciones al Software	150
D.3. Análisis de Planillas	156
E. Bandas de Frecuencia	157
F. Lecciones Aprendidas	159
F.1. Configuración de teléfonos comerciales	159
F.2. Correr un UE en RFSimulator con ruido blanco.	160
F.3. Uso de los logs	160
F.4. Otras Herramientas	161
F.5. ¡Ojo con las antenas!	162
F.6. Conexiones coaxiales	162
F.7. Conexión del PC con SDR X310	163
F.8. Conexión del PC con SDR B200	163
F.9. El problema del DNS	164
G. Glosario	165
G.1. Acrónimos	166
Referencias	169

Tabla de contenidos

Índice de tablas **173**

Índice de figuras **174**

Capítulo 1

Introducción

1.1. Motivación y Antecedentes

La quinta generación de redes móviles, comúnmente conocida como 5G, ha revolucionado el panorama de las comunicaciones a nivel mundial. Esta tecnología promete velocidades de conexión ultra rápidas, latencia mínima y una capacidad masiva para conectar dispositivos y aplicaciones en tiempo real. La 5G no solo es una evolución de las generaciones anteriores de redes móviles, sino que también es un habilitador fundamental para una variedad de aplicaciones transformadoras en campos como la Internet de las cosas (IoT), la salud, la industria y el entretenimiento.

El despliegue de las redes 5G ha generado una creciente demanda de investigación y desarrollo en el ámbito de las comunicaciones móviles. Los desafíos técnicos y las oportunidades de innovación son abundantes, y es importante contar con entornos de prueba donde los investigadores puedan experimentar con las tecnologías en ambientes controlados antes de su implementación en redes comerciales.

En este contexto, las maquetas, bancos de prueba o *test beds* desempeñan un papel fundamental al proporcionar un terreno de pruebas seguro y flexible para la experimentación con las capacidades y limitaciones de las redes 5G. Estos entornos permiten a los investigadores probar escenarios de uso, validar algoritmos y protocolos, y evaluar el rendimiento de dispositivos y sistemas en un entorno que simula las condiciones del mundo real.

El concepto de maqueta o modelo a escala es ampliamente utilizado, estudiado y conocido en todas las ramas de la industria, la ciencia y la creación de tecnología. En la industria de las telecomunicaciones y entre los operadores de redes, se usa el término para referirse a instalaciones de equipos de su línea de fabricación, o utilizados en su red, en facilidades propias tipo laboratorio. En estos entornos los equipos pueden ser medidos, configurados y manipulados en general, replicando situaciones reales sin afectar los sistemas en servicio. En este tipo de instalaciones los fabricantes prueban conceptos y prototipos de equipos y funcionalidades. Se realizan cursos de entrenamiento, se hacen pruebas de aceptación y se puede experimentar configuraciones novedosas antes de ponerlas en producción. En la-

Capítulo 1. Introducción

laboratorios de reparaciones se usan para ensayar placas o unidades posiblemente defectuosas o reparadas, y estudiar comportamientos anómalos. Los laboratorios de certificación las utilizan en los procesos de verificación de cumplimiento de normativas. En las instituciones académicas se usan como soporte de investigación tecnológica y como laboratorios de clase.

En los últimos veinte años, el software ha ido desplazando al hardware en la implementación de funciones de red. En muchas áreas, toda la implementación se realiza en software que se ejecuta sobre hardware genérico. El concepto de maqueta aplica de la misma forma a instalaciones de software o combinadas. El software de las redes móviles comerciales suele ser en su mayoría propietario y las licencias representan una parte significativa de los costos, tanto del sistema como de las maquetas. Para los actores que no implementan integralmente una red celular, como fabricantes de componentes, desarrolladores de software, integradores de sistemas o académicos, el costo de adquisición de estas licencias resulta difícil de asumir. Pero más importante que el costo, el software de los proveedores de sistemas no brinda la flexibilidad necesaria para su uso en investigación y desarrollo por ser propietario y cerrado. Por esta razón, comunidades de investigadores han creado sistemas de código abierto que tienden a cubrir estas necesidades.

En investigación y desarrollo de software se utilizan también los simuladores además de las maquetas. Un simulador es un programa que responde a una serie de entradas con las respuestas que daría el sistema simulado, sin ser ese sistema. Es un tipo de herramienta diferente y complementaria a la maqueta. Por ejemplo, un investigador desarrollando un algoritmo nuevo para una función cualquiera, puede conectarlo a un simulador, y obtener las respuestas que daría un sistema, con todas las variables o condiciones controladas. Eso facilita la prueba sistemática, en especial en las fases iniciales de un desarrollo. Por otro lado, en una maqueta que funciona realmente, es mucho más difícil tener controladas todas las condiciones, ya que muchas de ellas son respuestas a situaciones variables externas, físicas o lógicas. La maqueta provee un sistema de pruebas más complejo, pero próximo al funcionamiento real, donde no todas las respuestas posibles fueron preparadas por el desarrollador. Es común el uso combinado de ambas herramientas, para crear escenarios de pruebas complejos. Al igual que en las maquetas, existen simuladores comerciales y de código abierto para redes 5G, y para algunas de sus funciones.

Los sistemas de código abierto han tenido un impacto significativo en la evolución tecnológica al permitir a los desarrolladores acceder, modificar y compartir software de manera colaborativa y transparente. Dos de las plataformas más destacadas en el ámbito de las maquetas de redes móviles son OpenAirInterface (OAI) y Software Radio Systems Radio Access Network (srsRAN). OAI es una plataforma de código abierto dedicada al desarrollo de sistemas de comunicación móvil 4G y 5G. Esta plataforma proporciona un conjunto de herramientas, bibliotecas y paquetes de software para investigadores, ingenieros y operadores de redes que desean experimentar, desarrollar y probar tecnologías de comunicación avanzadas. OAI permite la creación de estaciones base, dispositivos de usuario y redes de núcleo 5G, corriendo sobre computadoras estándar. Utilizando equipos de radio definidos por software (SDR, por su sigla en inglés) adecuados, es posible imple-

1.2. Objetivo General

mentar redes celulares completas con esta tecnología. srsRAN es otra plataforma de código abierto que propone una solución equivalente para la Red de Acceso por Radio (RAN, por su sigla en inglés). Otras comunidades proponen soluciones para la red de núcleo de 5G.

En este trabajo estudiamos ambas soluciones e implementamos una maqueta de red 5G con la solución de OAI, con el propósito de dotar a la Facultad de Ingeniería de la UDELAR de una herramienta adicional de investigación, desarrollo y docencia. Este trabajo está inserto en programas de investigación de larga data. Se vincula a la amplia experiencia de trabajo con SDR y desarrollos en gnu-radio con los que se han recorrido los dos caminos. Es parte del proyecto de investigación “Herramientas de simulación de redes móviles de futura generación” (FVF-2021-128–DICYT). El simulador Py5chesSim, creado en la tesis de maestría de Gabriela Pereyra [31] y mejorado en trabajos posteriores, es un producto anterior de este proyecto de investigación que se completa con la presente maqueta y sus futuras posibles ampliaciones.

Este trabajo tiene el potencial de vincularse a otras líneas de investigación actuales. Por ejemplo en los que se han incorporado escenarios de uso de inteligencia artificial para optimizar la asignación de recursos de radio y para mejorar el funcionamiento de la red 5G. Estos incluyen el proyecto de investigación ANII “Inteligencia artificial aplicada a redes 5G”, en el marco del grupo ARTES (Análisis de Redes, Tráfico y Estadísticas de Servicio), así como el proyecto CSIC “Convergencia entre redes 5G/6G y redes ópticas: un enfoque holístico”, que abarca aspectos como la RAN, el transporte, el sincronismo y los orquestadores con el fin de lograr despliegues eficientes de redes con múltiples segmentos o *slices*.

1.2. Objetivo General

En este proyecto se busca implementar una Maqueta de Red 5G para fines de investigación y enseñanza en la Facultad de Ingeniería de la UDELAR. El objetivo es implementar una maqueta de una red 5G autónoma (*Stand Alone*) que abarque todas las funciones de emulación, incluyendo *Core Network* (CN), *Base Station* (BS) y *User Equipment* (UE), como se muestra en la figura 1.1.

Para lograr este propósito, se emplean herramientas preexistentes desarrolladas en código abierto y acceso gratuito. Se implementa la interfaz de radio mediante el uso de *Software Defined Radio* (SDR).

1.3. Objetivos Específicos

El objetivo general se desglosa en los siguientes objetivos específicos:

- Conocer las arquitecturas para las redes 5G.
- Comprender las soluciones basadas en srsRAN y OAI y determinar cuál de estas se ajusta mejor a los requisitos de investigación y desarrollo que hemos establecido.

Capítulo 1. Introducción

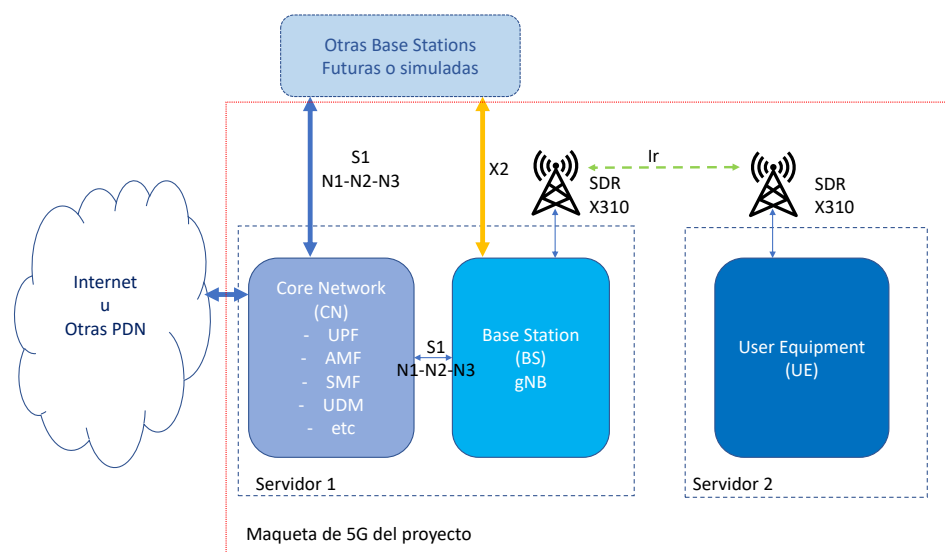


Figura 1.1: Diagrama general de la maqueta propuesta, que incluye Core Network y una radio base funcionando en una computadora con un SDR como interfaz de radio y un terminal de usuario en otra computadora, también con SDR.

- Realizar la documentación de la maqueta, en forma de manual de usuario.
- Caracterizar la maqueta, midiendo sus capacidades y detectando sus principales limitaciones.
- Implementar un desarrollo de software que valide sus potencialidades como base de desarrollo.

1.4. Alcance

El alcance de este proyecto implica la implementación de una red completa que incluye *Core Network* (CN) y *Base Station* (BS) en una misma máquina y *User Equipment* (UE) en otra máquina. Todas estas partes estarán conectadas a través de una interfaz de radio que utiliza dos *Software Defined Radios* (SDR). El objetivo es asegurar que el sistema básico esté operativo y disponga de todas las funciones necesarias para establecer una conexión de extremo a extremo a través de la red implementada.

Además, este proyecto contempla la creación de manuales de instalación y usuario que permitan replicar la instalación realizada, documenten la configuración y ajustes de todos los elementos de hardware y software utilizados, e instruyan sobre la creación de BS, UE y todos los tipos de servicios y casos de uso ensayados en este proyecto.

También se incluye en el alcance la implementación de un desarrollo de software que se integre al sistema para agregar una nueva funcionalidad. En este caso, hemos elegido trabajar en el *scheduler* de la radio base, con el propósito de demostrar

1.5. Productos y Contribuciones del proyecto

la viabilidad de implementar estos desarrollos en el sistema elegido. Y finalmente, otro objetivo opcional es agregar el uso de teléfonos comerciales como terminales de usuario.

1.5. Productos y Contribuciones del proyecto

Maqueta Funcionando

Hemos alcanzado con éxito la implementación de una maqueta de red 5G de extremo a extremo basada en OpenAirInterface (OAI). Se lograron cumplir todos los objetivos establecidos previamente para la implementación de la red. La maqueta demostró un rendimiento satisfactorio en términos de velocidad, capacidad y latencia, lo que indica que la solución es funcional y eficiente. La implementación fue sometida a una evaluación que incluyó pruebas de funcionamiento y rendimiento, y los resultados fueron satisfactorios en los aspectos evaluados. Entendemos que la red quedó validada técnica y operativamente, demostrando su capacidad para satisfacer las necesidades y requisitos planteados en una importante variedad de configuraciones. Se ensayaron además varias de las herramientas adicionales provistas por el sistema.

La incorporación de nuevas funcionalidades en el desarrollo de software aplicado al scheduler de la radio base confirma la idoneidad de esta plataforma como base de desarrollo. Se destaca que el cambio realizado quedó compilado con el conjunto del sistema, manteniéndose las características funcionales probadas. En resumen, estos aspectos destacan el éxito de la implementación de la maqueta y su capacidad para cumplir con los objetivos y requerimientos establecidos.

Manuales de Usuario

Los manuales de usuario proporcionan instrucciones detalladas que permiten replicar la implementación hecha. Allí se incluyen los diagramas de red de los escenarios ensayados, las configuraciones utilizadas y los parámetros indicados por comando en cada caso. Con los manuales realizados, se facilita la réplica del trabajo realizado y también se indica al usuario cómo modificarlos para nuevos casos de uso, siguiendo las normas de la 3GPP. Esto permite hacer configuraciones personalizadas para la ejecución de pruebas adaptadas a necesidades específicas. Los manuales y un resumen de las lecciones aprendidas se incluyen en los Apéndices A y F.

Desarrollo de un Scheduler Alternativo

El desarrollo de software de OpenAirInterface, implementado en lenguaje C, está organizado en forma jerárquica según la estructura del stack de protocolos. Esta estrategia simplifica considerablemente encontrar las áreas específicas del código donde se busca incorporar o modificar funcionalidades. Un aspecto clave de esta organización es la nomenclatura coherente que se aplica a las funciones implementadas en el código, en concordancia con las secciones de la normativa de la 3GPP.

Capítulo 1. Introducción

Dada esta organización, consideramos proporcionar una descripción indicativa de la estructura general del código. Además de esta estructura organizativa, realizamos un análisis del algoritmo utilizado para la asignación de recursos de radio, conocido como *proportional fair*. Como resultado de esta evaluación, desarrollamos un algoritmo alternativo. Esta implementación proporciona un sistema de asignación de recursos que permite manipular las políticas de asignación entre tres casos: *Proportional Fair*, *maxTBS* y *Round Robin*. Condiciones intermedias pueden ensayarse variando pesos por parámetros en la asignación. La integración cohesionada de este nuevo algoritmo proporciona la flexibilidad necesaria para manipular las políticas de asignación de recursos, permitiendo la incorporación de más casos de uso de manera eficiente.

Prueba de Equipos Comerciales

Utilizando SIMs programables y una grabadora adquiridos a Open-Cells Project, logramos configurar los parámetros necesarios para registrar con éxito dos dispositivos Samsung A33 5G en nuestra red. Como resultado de este proceso, una vez registrados, los teléfonos celulares muestran el nombre de la red como “Maq5GFing”. Estos dispositivos se conectan a Internet utilizando el acceso proporcionado por la maqueta. Además, hemos realizado verificaciones de alcance y cobertura, rendimiento y estabilidad con estos equipos. Validamos así el uso de esta red con celulares comerciales. El caso de éxito del uso de estos modelos con la red OAI fue reportado por nosotros en los foros de esta comunidad y en el registro que se lleva en la web de Open Cells Project.

Formación y Educación

La maqueta servirá como una herramienta valiosa para la formación y la educación en la Facultad de Ingeniería. Los estudiantes y profesionales tendrán la oportunidad de aprender sobre las redes 5G y la radio definida por software a través de la práctica y la experimentación en un entorno realista. Además, la maqueta facilitará la realización de experimentos y pruebas controladas, lo que ayudará a la investigación en redes 5G, incluyendo aspectos como el rendimiento de la red, la gestión de recursos y la optimización de protocolos.

Impacto en la Industria y Futuras Investigaciones

La capacidad de emular y probar sistemas 5G en una maqueta de código abierto puede ser de interés para fabricantes, operadores de redes y desarrolladores de soluciones en redes móviles. Este proyecto sienta las bases para investigaciones y desarrollos futuros en el campo de las comunicaciones móviles de próxima generación. Las áreas de investigación identificadas ofrecen oportunidades para continuar mejorando las capacidades de la maqueta y su aplicabilidad en una variedad de escenarios.

En resumen, la implementación de esta maqueta de red 5G contribuirá al avance de las comunicaciones móviles y la investigación en la Facultad de Ingeniería.

Al proporcionar una plataforma de experimentación, formación y desarrollo basada en código abierto, este proyecto impulsa la innovación y la comprensión de las tecnologías de redes móviles en un contexto académico y de investigación.

1.6. Organización del Documento

El resto del documento está organizado en los siguientes capítulos:

En el Capítulo 2 se proporciona un panorama general de los fundamentos de las redes 5G, su evolución y los desafíos que enfrentan. Además se incluye un conjunto de notas técnicas con conceptos claves para la implementación.

En el Capítulo 3 se identifica la plataforma que mejor se adapta a los objetivos de investigación y desarrollo fundamentando la selección.

En el Capítulo 4 se presenta el hardware mínimo necesario para desplegar la red, se identifican posibles líneas de investigación futura, se detallan las opciones de configuración utilizadas, se describen los escenarios de uso y se presenta el presupuesto total de la maqueta.

En el Capítulo 5 se reportan el diseño de pruebas en diferentes escenarios para caracterizar los límites y las capacidades de la maqueta, y se presentan los resultados en gráficos comparativos.

En el Capítulo 6 se selecciona la aplicación a desarrollar, en este caso, el *scheduler* de la radio base. Se describe la estructura general del código y se presenta un algoritmo de *scheduling* alternativo que se integra a la plataforma. A continuación, se reportan los resultados obtenidos al ejecutar las mismas configuraciones utilizadas para caracterizar la maqueta, pero esta vez utilizando el algoritmo implementado. Estos resultados se presentan en tablas y gráficos comparativos para su análisis y evaluación.

Finalmente, en el Capítulo 7 se encuentran las conclusiones finales del proyecto y se identifican algunas posibles líneas de trabajo futuras para su continuidad.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

Redes Móviles 5G

Este capítulo proporciona una visión general de los fundamentos y las metas de las redes móviles de quinta generación. Se busca entender los motivos que la generaron, las ideas clave de su implementación y sus características esenciales. Se profundiza en los aspectos relevantes para la comprensión de este trabajo. En este sentido se resalta la importancia de la evolución continua, la adaptación a nuevos casos de uso y los desafíos en el futuro, que requerirán el uso de maquetas como esta para su investigación y desarrollo.

El tema es extenso y complejo, con una gran cantidad de definiciones y siglas. Su uso es necesario para lograr precisión en el documento. Se incluye un glosario en el Apéndice G para ayudar en la lectura.

2.1. ¿Qué es 5G?

Más velocidad de datos, menos latencia, más confiabilidad y muchísimos más dispositivos conectados por metro cuadrado, sumados a formas de funcionamiento con mayor ahorro de energía, son los objetivos y las promesas de la quinta generación de redes móviles. Esas características habilitan anchos de banda inalámbrico de Gb/s en móviles, pero también en servicios fijos que compiten con el acceso por fibra óptica. Permite internet de las cosas en mucho mayor escala, aplicaciones de vehículos autónomos con comunicaciones “vehicle-to-everything” (V2X), automatización e industria 4.0, video de mayor calidad ininterrumpido, realidad virtual y muchas otras aplicaciones demandantes. Para entender los desafíos y las opciones que se toman y se tomarán en los estándares de esta tecnología, hay que tener presente la diversidad de objetivos que se persiguen.

5G surge como una evolución de LTE, con mucho cuidado de la compatibilidad y de los escenarios de evolución entre ambas generaciones. Todos los casos de uso que inspiran esta evolución, están ya atendidos parcialmente por LTE. Entonces, ¿cuál es la necesidad del cambio? LTE fue diseñada como una red de paquetes que brinda el mismo servicio a todos sus usuarios, independientemente de la aplicación. Este servicio está orientado a la banda ancha móvil y optimizado para ella, que es su uso más masivo. Pero el diseño de la interfaz de radio penaliza con latencias muy

Capítulo 2. Redes Móviles 5G

altas para algunos usos y demanda conexiones cuidadas que requieren alto consumo de energía para otros casos. Eso, sumado al continuo crecimiento de la demanda de datos, que ha sido atendido hasta ahora agregando portadoras en nuevas bandas de frecuencia para LTE, impulsa la necesidad de cambios más de fondo, que permitan afrontar ese crecimiento a futuro y que pueda brindar soluciones con prestaciones diferentes, a demandas diferentes.

El 3rd Generation Partnership Project (3GPP) define la primera arquitectura completa de 5G en la *Release 15*. Esta institución es actualmente el único organismo internacional relevante en la normalización de sistemas móviles y es el responsable de la normalización de 3G, 4G y ahora 5G.

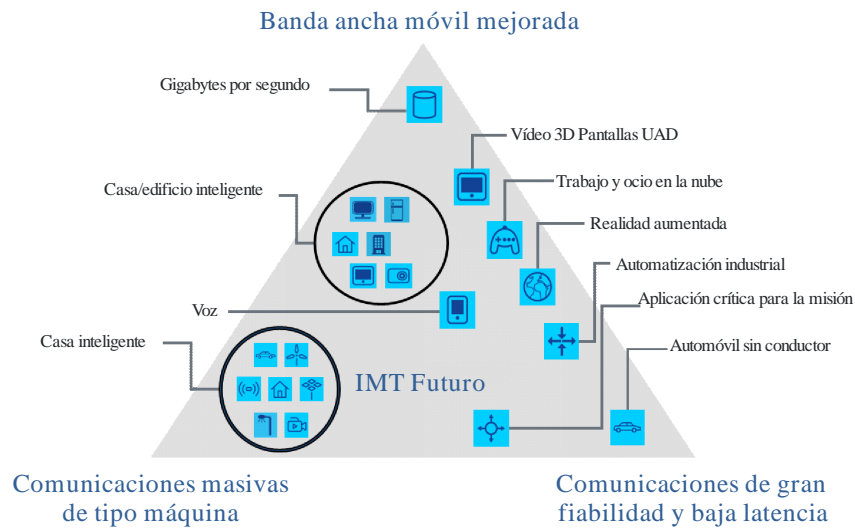
La Unión Internacional de Telecomunicaciones (ITU) en la Recomendación M.2083-0 [23] define los primeros tres casos de uso hacia los que se orienta esta tecnología.

- **Enhanced Mobile BroadBand (eMBB):** Este caso define las velocidades de datos de pico y promedio, capacidad y cobertura que debe alcanzar un diseño 5G, en comparación con los alcanzados en las redes anteriores. Se definen objetivos de *Down Link* (DL) de 20 Gb/s y de *Up Link* (UL) de 10 Gb/s.
- **Massive Machine-Type Communication (mMTC):** Soporte 5G para dispositivos de internet de las cosas (*Internet Of Things* IoT). Cubre tanto los casos de dispositivos con baja tasa de datos, poco ancho de banda, que requieren solo algunas ráfagas de transmisiones y esquemas de máximo ahorro de baterías, como casos de muy alta tasa de datos sobre un gran ancho de banda.
- **Ultra-Reliable and Low-Latency Communications (URLLC):** Caso enfocado en servicios que requieren comunicaciones en tiempo real y/o libres de fallas. Ejemplos: Vehículos autónomos, internet industrial, protección de infraestructura, *Smart Grids*, etc.

Estos casos de uso son una clasificación arbitraria para organizar el trabajo de investigación y especificación. En los hechos, muchos casos de aplicación no serán exactamente esos, podrían por ejemplo requerir confiabilidad y no baja latencia, o baja latencia y alto ancho de banda, o cualquier combinación entre ellos como se ejemplifica en la figura 2.1.

New Radio (NR) es la interfaz de radio definida en la serie 38 de especificaciones del 3GPP. NR es a 5G lo que LTE a 4G. En ambos casos las siglas nombran la interfaz de radio, aunque popularmente se utilicen como sinónimos de la tecnología. Si bien NR se inicia reutilizando las mismas estructuras definidas en LTE, al ser una nueva generación se libera de la obligación de ser compatible para atrás, permitiendo cambios disruptivos. En ella se define *Orthogonal Frequency-Division Multiplexing* (OFDM) como modulación de *Down Link* (DL) y de *Up Link* (UL) [1], lo que facilita la conexión directa entre equipos de usuarios, que se incluye en la *Release 17*. También autoriza el uso opcional de *Discrete Fourier Transform*

2.2. Estrategias de Despliegue



M.2083-02

Figura 2.1: Ejemplos de casos de aplicación de 5G, según la clasificación propuesta por la ITU en la Recomendación M.2083-0 [23].

spread OFDM (DFT-OFDM) en el UL, lo que permite la compatibilidad con LTE, y la reducción de consumo en el UE que lo utilice.

Pero 5G es más que NR, es un Core Network (CN) diseñado nativamente para *cloud computing* como estrategia de evolución ágil y dinámica. Incluye la división lógica de las radio bases, ahora llamadas *new generation Node B* (gNB), en *Central Units* (CU) y *Distributed Units* (DU), habilitando la virtualización en partes de la RAN. Es la posibilidad de agregar Inteligencia Artificial (AI) y Aprendizaje Automático (ML) como mecanismos de análisis y optimización de la red y el uso de sus recursos. Es *Edge Computing* como facilitador para servicios que requieran baja latencia, tales como juegos en red, realidad virtual y realidad aumentada. Es multiplicar varias veces los dispositivos conectados y es poder atender todos los escenarios diferentes, con la flexibilidad necesaria para adaptarse a los requerimientos de cada uno. Es adaptar los servicios disponibles a las realidades y necesidades de mercados diferentes.

2.2. Estrategias de Despliegue

Dos modalidades de despliegue se han previsto desde la norma y se están implementando: *Non Stand Alone* (NSA) y *Stand Alone* (SA). En NSA se instalan radio bases 5G (gNB) insertas en una red LTE, o se mantienen radiobases LTE (eNB) en una red 5G ya implementada. En la modalidad SA las redes se definen en forma independiente, con todas las radio bases de la misma tecnología. En la modalidad SA todas las funcionalidades definidas en 5G son implementadas. La

Capítulo 2. Redes Móviles 5G

modalidad NSA, en sus diferentes versiones permite despliegues rápidos, utilizando la cobertura y red de núcleo de la red 4G con agregados de 5G aumentando la capacidad de la red mientras se alcanzan coberturas importantes, acompañando la inversión al despliegue de la red.

En el trabajo previo a la primera estandarización de 5G se analizaron los ocho escenarios posibles, las cuatro combinaciones SA, donde gNB o eNB se conectan a los Core Network de 4G o de 5G y las cuatro combinaciones NSA, donde una RAN con ambos tipos de radiobases, pero con la señalización manejada por 4G, o con la señalización manejada por 5G se conectan a Core Networks de ambas tecnologías [19]. En la figura 2.2 se representan esos escenarios. Los casos 6, donde solo hay gNB conectados a *Enhanced Packet Core* (EPC) de 4G como red de núcleo y 8, donde una RAN con eNB y gNB, con señalización 5G, se conecta a un EPC nunca fueron normalizados por no tener utilidad práctica. No es razonable la inversión de cambiar la RAN y no utilizar esas ventajas por no cambiar el Core Network.

La opción 3 de la figura fue la primer arquitectura normalizada en la Release 15 del 3GPP, la primer versión del conjunto de normas de 5G. Allí sigue utilizando el *Enhanced Packet Core* (EPC) de 4G como red de núcleo y se maneja la señalización de los UE mediante una celda de una radio base 4G (*enhanced Node B*, eNB) que llegue al UE, y se la identifica como ancla. El eNB actúa como celda primaria (PCell) en un esquema de múltiple conexión, con el UE conectado a un grupo de celdas independientes en forma simultánea. Pueden ser celdas de la misma radio base formando un grupo de Carrier Agreggation (CA) o de distintas radio bases en modo Multiple Connection. El sistema LTE debe actualizarse para permitir estas conexiones entre distintas Tecnologías de Acceso por Radio (RAT por su sigla en inglés). La celda 5G actúa como celda secundaria (SCell), con el único rol de aumentar la capacidad de transmisión de datos. La opción 1, corresponde a la arquitectura previa de la red 4G. Las opciones 2, 4, 5, 7 y algunas variantes se han ido agregando en distintas actualizaciones de la normativa. Estas opciones permiten elegir entre varios escenarios de transición.

La modalidad *Stand Alone* corresponde a redes independientes. La opción 2 es una red 5G completa, con *Core Network 5G* implementado e independiente de la red 4G. La maqueta que se implementa en este proyecto sigue esta opción, aunque tiene la capacidad de implementar todos los escenarios de la figura 2.2, lo que puede estudiarse a futuro. Los despliegues comerciales actuales en Uruguay, siguen por ahora las arquitecturas NSA, según la opción 3x.

2.3. Características Principales

Repasaremos las principales características de NR en función de cómo aportan a los objetivos iniciales: mayor velocidad de transferencia de datos, menor latencia, mayor flexibilidad y alta confiabilidad. El hilo conductor es ver qué se hace y para qué, dejando los detalles del cómo a la literatura especializada [13] [21]. Solo profundizaremos en los aspectos que se requieren para la comprensión de este trabajo.

2.3. Características Principales

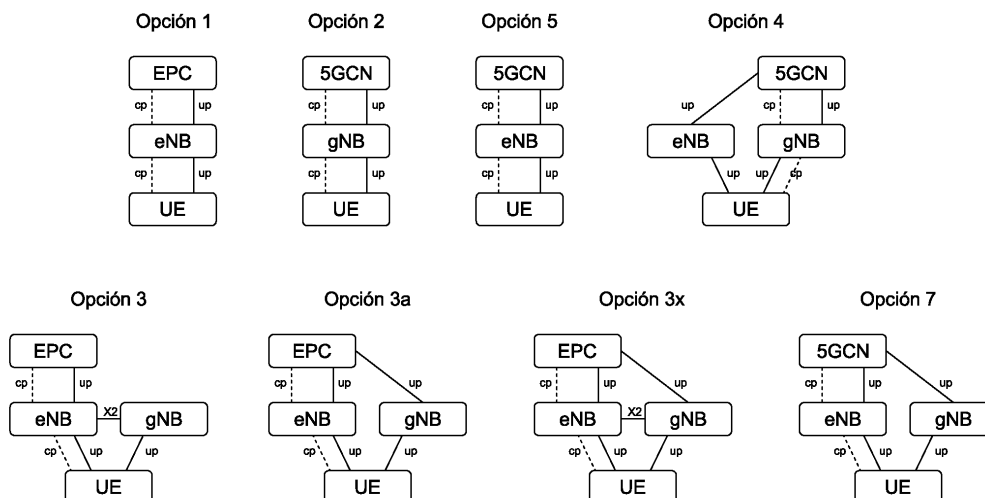


Figura 2.2: Distintas formas de conexión de los planos de usuario y de control en despliegues Non Stand Alone con las redes 4G y 5G interdependientes y Stand Alone con las redes 4G y 5G completamente separadas. En este proyecto se implementa la opción 2, 5G SA con la tecnología NR completamente en la RAN y el core 5G. En las demás opciones se utiliza LTE ya sea en el plano de control o de usuario o ambos en combinación con NR.

Mayor Throughput

Desde la *Release 15* inicial, se incluye en 5G la posibilidad de uso de grandes rangos del espectro. Se define el Frequency Range 1 (FR1) con las frecuencia por debajo de 6 GHz y el Frequency Range 2 (FR2) de 24.25 GHz a 52.6 GHz. En las versiones posteriores se estudia el agregado de bandas adicionales, extendiendo ambos bloques. Las bandas de FR1 incluyen todas las bandas utilizadas en generaciones anteriores. Utilizan la misma nomenclatura definida para LTE, simplemente agregando el prefijo n, por ejemplo, la banda 28 de 700 MHz LTE es n28 en NR 700 MHz.

El uso del espectro radioeléctrico se define en base a asignaciones de alcance nacional. Los organismos técnicos internacionales hacen recomendaciones. La producción industrial de dispositivos impulsa formas culturales de su uso. Pero la decisión final es hecha por entes reguladores con potestades jurídicas sobre territorios nacionales. El espectro es un recurso demandado para múltiples usos, público o privado, civil o militar, de servicios o comerciales, por lo que las asignaciones conllevan importantes conflictos de interés. Estas definiciones se toman en momentos históricos y tecnológicos distintos, con plazos de validez generalmente largos. En consecuencia, soluciones distintas se aplican en países diferentes. Varios organismos internacionales y regionales hacen esfuerzos por unificar criterios con resultados parciales.

El International Telecommunication Union, Radiocommunication Sector (ITU-R) estandariza y clasifica todas las bandas posibles, definiendo los rangos de frecuencias, usos y canalizaciones. En especial, para el uso en redes móviles, es importante obtener bandas con consenso mundial, para que un viajero pueda utilizar

Capítulo 2. Redes Móviles 5G

su dispositivo en todos lados, y para que la fabricación se haga a escala global, permitiendo economías que ayudan a la difusión de los sistemas. En el Apéndice E se incluyen dos tablas con las bandas definidas para el uso en 5G, en los dos rangos ya nombrados y cuyo uso discutimos a continuación.

En FR1 se definen anchos de banda de hasta 100 MHz y se permite el uso de bandas definidas para *Frequency Domain Duplexing* (FDD), un sistema en el que se utilizan distintas frecuencias para transmitir y recibir. También se definen bandas donde se autoriza el uso de *Time Domain Duplexing* (TDD), sistema en el que se transmite y recibe en la misma frecuencia, en instantes de tiempo diferentes. Se han ampliado en algunas regiones las bandas bajas asignadas, permitiendo alcances mayores y buena penetración. Es el caso de T-Mobile en Estados Unidos que ofrece servicio 5G en la banda n71 de 600 MHz. También se agrega el uso de bandas medias, de 3 GHz a 7 GHz, que no eran utilizadas por las tecnologías anteriores. Esto permite la adjudicación de bloques de 100 MHz o más por operador, dependiendo del regulador y la disponibilidad en cada país. La posibilidad de uso es resultado del esfuerzo realizado por liberar la banda C del uso para transmisiones satelitales, de la menor demanda de enlaces de microondas de larga distancia como sistema de transporte y de la reducción del uso de redes de acceso inalámbrico fijo. La banda de 6 GHz fue declarada como no licenciada en Estados Unidos, con un uso potencial esperado en Wi-Fi. En Europa se asignó 6L (5925 MHz - 6425 MHz) para uso no licenciado, reservando 6U (6425 MHz-7125 MHz) como banda para *International Mobile Telecommunications* (IMT) licenciadas. En China se reserva toda banda 6 GHz licenciada. América Latina tiene esta discusión regulatoria activa en varios países. En Uruguay fue ampliamente utilizada en el pasado para enlaces troncales de microondas. Su uso futuro no tiene una definición tomada aún. En este proyecto se utilizaron frecuencias de las bandas n66, n41, n77 y n78. En la banda n78 están las frecuencias asignadas para 5G en nuestro país.

En FR2 se permiten anchos de banda de hasta 400 MHz utilizando TDD. Se permite el agregado de hasta 16 portadoras, con un ancho de banda máximo acumulado de 800 MHz para un usuario. Si comparamos que en LTE Advanced (LTE-A, LTE con las mejoras introducidas por *Release 10* y posteriores) el ancho de banda máximo por portadora es de 20 MHz y a lo sumo se pueden agregar 5 portadoras con un ancho de banda agregado de 100 MHz, tenemos aquí la primera explicación del mayor *throughput*, usando la técnica conocida como *Carrier Aggregation* (CA).

En las bandas medias y altas la mayor frecuencia impone una mayor atenuación. Eso se mitiga aumentando la ganancia de la señal en la dirección del usuario, mediante *beamforming*. Esta técnica implica tener un arreglo de antenas y coordinando las fases y amplitudes de la señal en cada elemento, se puede concentrar el haz en la dirección deseada. Cuando la variación de amplitud y fase se hace en el procesamiento digital y la conversión analógico digital se hace directamente a la salida, la técnica se denomina *beamforming* digital. La otra técnica utilizada consiste en alterar la amplitud y fase sobre la señal de radiofrecuencia entre el transmisor y cada elemento de la antena. Se la conoce como *beamforming* analógico. El *beamforming* digital requiere un conversor A/D por cada salida de radio,

2.3. Características Principales

y una salida de radio por cada elemento de antena. Tiene como ventaja, que es posible hacer multiplexación espacial, enviando señales en el mismo tiempo y frecuencia a dispositivos que estén suficientemente separados. Para esto se utilizan matrices de pre codificación para combinar linealmente las señales a multiplexar de manera conveniente, para ser recuperadas en el receptor en forma similar al envío de múltiples señales a un mismo dispositivo usando *Multiple Input Multiple Output* (MIMO). Esta técnica se conoce como *massive* MIMO. La multiplexación espacial también es causa de aumento del *throughput*.

En el *beamforming* analógico, sólo se puede enviar una única señal, debido a que es originada en un solo puerto de RF. Pero para las frecuencias mas altas, es la única técnica disponible por ahora, aunque nuevos desarrollos como *beamforming* híbrido (HBF), combinando ambas técnicas, antes de las etapas de frecuencia intermedia, presentan avances continuamente.

Desde la *Release 16*, se habilita el uso de NR en bandas de frecuencias no licenciadas, lo que aporta a mayor capacidad de transferencia de datos. Se puede utilizar como portadoras agregadas a celdas principales en bandas licenciadas que garantizan la conectividad, para aumentar el ancho de banda disponible, como *Dual Connectivity* con celdas primarias NR o LTE, o también como celdas independientes conectadas al CN5G funcionando exclusivamente en ese espectro [4]. Se conoce como *Dual Connectivity* a brindar servicio a un móvil desde dos radio bases diferentes. Este desarrollo está inserto en la discusión ya mencionada sobre el uso del espectro de 6 GHz [20].

Menor Latencia

A diferencia de LTE, NR permite variar el espaciamiento entre subportadoras (SCS, por su sigla en inglés). Mientras que en LTE está fijo en 15 kHz, en NR es $2^n \times 15$ KHz, con $n \in [0, 4]$ en la *Release 15* y extendiéndose hasta 6 en la *Release 17*. En FR1 se pueden usar separaciones de 15 kHz, 30 kHz y 60 kHz y son las alternativas utilizadas en la maqueta de este trabajo. En FR2 se utilizan 60 kHz, 120 kHz y 240 kHz como separación entre subportadoras. En la *Release 17* se agregan para FR2 los espaciamientos de 480 kHz y 960 kHz.

Al aumentar el ancho de banda de cada subportadora se baja el tiempo de duración del símbolo. El tiempo de símbolo debe ser el inverso de la separación entre subportadoras para asegurar la ortogonalidad entre ellas. La estructura de trama de radio en NR es muy parecida a la de LTE. Una trama dura 10 ms. Está dividida en 10 subtramas de 1 ms, y cada subtrama se divide en slots de 14 símbolos OFDM cada uno. La cantidad de slots por subtrama depende de la separación de portadoras. Para 15 kHz hay un slot por subtrama y la estructura de trama queda idéntica a la de LTE. Esto garantiza la compatibilidad entre ambas redes y permite la convivencia de las dos interfaces superpuestas en una misma celda, en el modo *Spectrum Sharing*. Espaciamientos mayores reducen la cantidad de subportadoras para un ancho de banda dado, pero también reducen el tiempo de símbolo, reduciendo la latencia para una misma capacidad de transferencia. Mas detalles en relación a OFDM, numerología y estructura de trama se encuentran en la sección siguiente.

Capítulo 2. Redes Móviles 5G

El uso de una subestructura llamada minislot permite mejoras adicionales importantes en latencia. Mientras que en LTE cada transmisión de datos requiere esperar el inicio del siguiente slot, en NR puede comenzar a transmitirse inmediatamente, usando fracciones de slots de 2, 4 o 7 símbolos OFDM. El comienzo de transmisión inmediato, es un recurso adicional útil también en bandas no licenciadas. En estas bandas es necesario verificar la disponibilidad del canal antes de transmitir y se requiere comenzar de inmediato una vez sensado, para no perder la ventana de oportunidad.

Las mejoras en los tiempos de procesamiento de los protocolos, la mayor restricción de tiempo para enviar reconocimientos, y la concentración de la información de control al inicio de cada slot que permite procesar la recepción a medida de que va llegando sin necesidad de almacenarla toda en memoria previamente, también inciden en la menor latencia. En un plano más arriba, dos técnicas hacen uso de estas ventajas para lograr latencias bajas: el *puncturing*, que consiste en interrumpir transmisiones largas si llegan mensajes marcados con prioridad y requerimiento de servicio de baja latencia y el *network slicing*, que implica dividir transversalmente la red en porciones que atraviesan todos los componentes, y que cada una tiene políticas de asignación de recursos específica para el tipo de servicio para la que está diseñada.

Flexibilidad

Una innovación de NR son las partes de ancho de banda, o *Bandwidth Parts* (BWP). El ancho de banda de una portadora puede dividirse en partes con diferente tamaño, diferente numerología y señalización independiente. Eso permite atender dispositivos de usuario con capacidades diferentes, sin que deban ser capaces de manejar todo el ancho de banda disponible. Permite variar el ancho de banda en uso según la demanda de requerimientos, operando con menor consumo de energía y de recursos cuando no se requieren. También permite establecer capas de servicios diferentes sobre la misma portadora con formas de asignación de recursos diferenciadas.

Para atender la demanda de servicios diferentes es posible dividir la red 5G en segmentos transversales a la RAN, el *Core Network* y los sistemas de transporte vinculados llamados *slices*. Cada uno administrará los servicios que brinda a los usuarios finales con políticas diferentes, atendiendo los objetivos de cada servicio. A partir de configuraciones propias, con asignación de recursos a nivel macro entre *slices* y con administración de los recursos de cada *slice* según sus propias definiciones, se permite a la red a funcionar como un conjunto de redes independientes superpuestas, sobre los mismos recursos de infraestructura y espectro. El *Network Slicing* así definido, es una herramienta para construir servicios diferenciados según casos de aplicación, redes virtuales, para compartir entre operadores diferentes o redes privadas virtuales enfocadas a mercados empresariales verticales, todo sobre la misma infraestructura física. Es posible pensarlo como calidades de servicio, solo que no segmentan un único conjunto de recursos de un canal o enlace en un mismo protocolo, sino que lo hacen sobre un ecosistema completo de recursos físicos y lógicos distintos, con administraciones independientes, para lograr objetivos en

2.4. Conceptos Claves para la Implementación

forma conjunta.

Confiabilidad

Con técnicas de *Dual Connectivity* o de *Carrier Aggregation*, es posible combinar el uso de un canal de menor frecuencia, y por tanto de más alcance y penetración, con otra portadora de alta frecuencia, con peores condiciones de propagación pero mucho ancho de banda. Esta estrategia permite combinar alto tráfico con una conexión confiable, ya que permite descargar de trabajo a la portadora más baja, traficando por la alta en todos los casos posibles. También es posible hacerlo combinando portadoras LTE con otras NR, lo que facilita el reuso de la infraestructura existente para esta característica.

La variación de la separación entre subportadoras permite variar la duración del slot, lo que permite opciones de optimización de latencia, velocidad de transferencia o de disponibilidad para distintos niveles de servicios. Espaciamientos de frecuencia más grandes ayudan a generar formas de onda más robustas, lo que es necesario a frecuencias altas donde el manejo del ruido de fase es más complicado. En los sistemas OFDM el prefijo cíclico (CP) ayuda a mitigar la interferencia intersimbólica (ISI) causada por los diferentes retardos en el canal. En NR, debido a las variaciones permitidas en el espaciamiento de subportadoras, el largo del CP debe variar adaptándose a la configuración. El uso de numerologías más altas acorta el tiempo de símbolo, y con ello se acorta el tiempo del prefijo cíclico, reduciendo su eficacia ante el ISI. El uso de frecuencias altas con menor propagación, reduce el área de cobertura acotando el problema. El uso de *beamforming* mejora la cobertura pero también limita el universo de caminos múltiples posibles, reduciendo el retardo máximo posible.

El CSI (*Channel State Information*), además de usarse para ecualización, mediciones del estado de canal y MIMO clásico, agrega en 5G NR un juego de datos y mediciones para controlar el *beamforming*, permitiendo el control dinámico de distintos haces, optimizándolos para usuarios individuales específicos. Cuanto mejor y más precisa es la información de CSI, mejor es la adaptación del enlace.

2.4. Conceptos Claves para la Implementación

En esta sección, se exploran algunos conceptos técnicos necesarios para comprender y ejecutar con éxito la implementación de la maqueta.

1- Duplexado por División en Tiempo y Frecuencia (TDD/FDD): Examina la diferencia entre TDD y FDD, dos enfoques clave en la asignación de espectro en redes 5G y cómo se aplican en la transmisión de datos ascendentes y descendentes. Son conceptos necesarios a la hora de configurar la frecuencia de trabajo de la maqueta, y de predecir su rendimiento esperado.

2- OFDM (Multiplexación por División en Frecuencia Ortogonal), Numerología y Estructura de Trama en 5G : Explora las características principales de OFDM y cómo ha evolucionado para convertirse en la base de la modulación en 5G, desempeñando un papel crucial en la eficiencia espectral. A

Capítulo 2. Redes Móviles 5G

continuación, se describe la Numerología y la Estructura de Trama en 5G, donde se examina cómo se estructuran los recursos de la red, incluyendo los conceptos de trama, slots, símbolos y subportadoras. También se requiere manejar estos conceptos para configurar el ancho de banda y estimar la capacidad del sistema configurado.

3- Sincronización: Analiza el proceso de sincronización entre UE y gNB. Solo entendiendo este proceso se puede diagnosticar los problemas de falla inicial de la conexión. La maqueta requiere que se configure correctamente la ubicación del bloque de sincronismo, y que sea buscado de esa forma por el UE por software.

4- Tamaño del Bloque de Transporte (TBS): Considera los factores que afectan el TBS y como se calcula en el caso de NR, para maximizar la eficiencia del sistema. La capacidad de transmisión en Bytes de cada ventana de oportunidad para un UE dado, es un concepto central en el manejo de la capa dos y en el desarrollo de software abordado.

5- HARQ (Hybrid Automatic Repeat Request) y Retransmisiones Compartidas entre Capa Dos y Capa Física : Aborda la técnica de retransmisión HARQ y cómo se utiliza para mejorar la confiabilidad de las comunicaciones en 5G. Interpretar su funcionamiento y comprender el por qué de sus incidencias es de las principales herramientas de diagnóstico con que se cuenta al trabajar con un sistema que no incluye un software de análisis que extraiga y presente los índices de performance en forma amigable.

2.4.1. Duplexado por División en Tiempo y Frecuencia

El duplexado por división en tiempo y frecuencia (TDD y FDD, por sus siglas en inglés) son dos técnicas de la comunicación inalámbrica que permiten la transmisión bidireccional de datos en las redes de comunicaciones en general. Son técnicas para compartir el uso del medio de transmisión, que es común a todos los usuarios. Son el modo de garantizar una comunicación eficiente y confiable, con niveles de interferencia aceptables en diferentes tipos de redes, incluyendo las redes móviles, y la red 5G en particular.

La figura 2.3 ilustra la diferencia entre FDD y TDD.

Duplexado por División de Frecuencias (FDD)

El FDD asigna diferentes bandas de frecuencia para la transmisión y la recepción de datos. En un sistema FDD, las comunicaciones de *upstream* (del dispositivo al punto de acceso) y *downstream* (del punto de acceso al dispositivo) utilizan bandas de frecuencia separadas. Las señales de transmisión y recepción no se superponen en el dominio de la frecuencia, lo que reduce las interferencias entre ellas.

FDD se ha utilizado ampliamente en tecnologías de comunicación móvil anteriores, como 3G y 4G LTE. Tiene la ventaja de la simplicidad, ya que cada sentido de la comunicación se resuelve en forma independiente. Sin embargo, una de las limitaciones es que requiere el doble de espectro de frecuencia para funcionar, lo que impacta en el aumento del costo del sistema. Otra complejidad técnica es que

2.4. Conceptos Claves para la Implementación

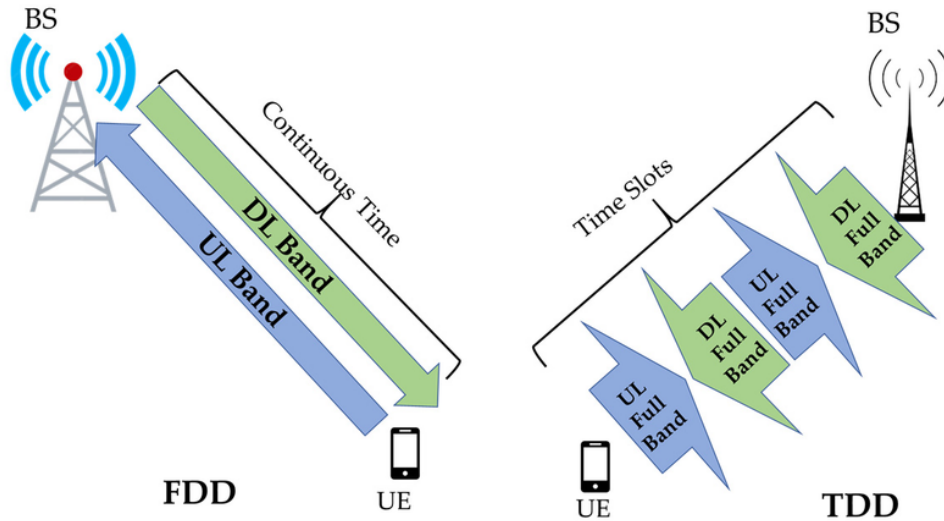


Figura 2.3: Duplexado en tiempo y frecuencia [36].

el canal no es simétrico, ya que la respuesta del canal depende de la frecuencia. Esto impone a los sistemas FDD dos evaluaciones independientes del canal, una en cada sentido.

Las bandas de frecuencias normalizadas por el ITU-T especifican si se usan para FDD o para TDD, ya que las FDD requieren dos rangos iguales y apareados. Todos los subrangos que se asignan a operadores dentro de una banda dada, tienen la misma separación entre las portadoras de UL y de DL. Como las frecuencias más bajas tienen mayor alcance, se reservan las frecuencias menores para la transmisión de los dispositivos de usuario que transmiten con menor potencia.

Duplexado por División en el Tiempo (TDD)

El TDD es una técnica en la que el mismo espectro de frecuencia se comparte entre la transmisión y la recepción, pero se realiza en diferentes intervalos de tiempo. En un sistema TDD, un dispositivo alterna entre transmitir y recibir en intervalos de tiempo separados, lo que permite el uso eficiente del espectro de frecuencia disponible. En algunas redes inalámbricas se utilizan técnicas de acceso aleatorio y control de colisiones, como en Wi-Fi, y en otras un nodo con funciones maestras distribuye el uso de ranuras de tiempo entre los usuarios que lo requieren, como en las redes de acceso por satélite V-SAT. En las redes móviles 5G, el tiempo de transmisión de las radio bases y de los usuarios está pautado por las radio bases. Se asigna un número de *slots* de tiempo para la transmisión del gNB y otros para los usuarios. El plano de control le indica a cada usuario en qué *slots* y en qué frecuencias le corresponde transmitir.

El TDD requiere que todos los gNB estén correctamente sincronizados, para que dentro de un área transmitan en el mismo momento. Como sus transmisores son más potentes que los de los UE, si no se sincronizan las ventanas de transmisión

Capítulo 2. Redes Móviles 5G

y de recepción, los gNB cercanos impedirán recibir a los UE ubicados en la celda. Otra complejidad es que se requiere una ventana de tiempo de espera entre la ventana de recepción en el gNB y la de transmisión. Esto es debido a que los distintos UE no estarán sincronizados perfectamente y si la última transmisión está activa cuando se empieza a transmitir por el gNB, será ininteligible. Los tiempos de conmutación de la electrónica entre transmisión y recepción son de menor orden, pero existen e imponen un mínimo a respetar.

Las ventajas de TDD están en el uso de la misma frecuencia en ambos sentidos. Esto hace simétrico al canal, por lo que se requiere un menor esfuerzo de evaluación. Esto es relevante en el uso de *beamforming*, que se explica más adelante, ya que las condiciones detectadas en recepción se pueden usar en transmisión, sin necesidad de un lazo de control más lento y con más uso de recursos. La misma razón es más relevante en las frecuencias altas de FR2, donde además se usan anchos de banda muy grandes, que dejarían muy separadas las dos portadoras con respuestas muy diferentes.

Una de las ventajas del TDD es la posibilidad de variar dinámicamente la asignación de slots de UL y de DL. Este modo de uso es nuevo, y hay menos experiencias, pero ha mostrado ser eficaz en entornos de acceso para usuarios fijos. Para el uso de TDD dinámico en redes móviles se requiere que esos cambios sean coordinados entre celdas vecinas, lo que exige técnicas de optimización aún en desarrollo. Esto le daría flexibilidad para adaptarse a las demandas cambiantes de tráfico de datos. En este proyecto se implementan gNB TDD en casi todos los casos de ensayo, siempre con asignación estática de los slots.

Aplicaciones en Redes 5G

En las redes 5G, tanto el TDD como el FDD se utilizan en diferentes escenarios y bandas de frecuencia. TDD se utiliza comúnmente en las bandas de frecuencia medias y altas, como las bandas milimétricas (mmWave). Esto es particularmente útil para aplicaciones de alta velocidad y baja latencia, como la transmisión de video en tiempo real y la realidad virtual/aumentada.

Por otro lado, FDD se utiliza en bandas de frecuencia sub-6 GHz en 5G, especialmente en escenarios de despliegue de cobertura amplia y de reutilización de rangos de espectro ya asignados a los operadores.

En resumen, el duplexado en tiempo y frecuencia son técnicas que permiten una comunicación eficiente y flexible, adaptándose a una variedad de aplicaciones y requisitos de tráfico de datos. Estas técnicas desempeñan un papel crucial en la capacidad de las redes 5G para ofrecer velocidades ultrarrápidas, baja latencia y conectividad confiable.

2.4.2. OFDM, Numerología y Estructura de Trama en 5G

La interfaz de radio de 5G utiliza OFDM, al igual que el DL en LTE. Es una forma de modulación que permite manejar anchos de banda grandes en forma eficiente. Consiste en dividir el espectro en una gran cantidad de subportadoras, que por tener un espectro angosto pueden considerarse constantes en su respuesta

2.4. Conceptos Claves para la Implementación

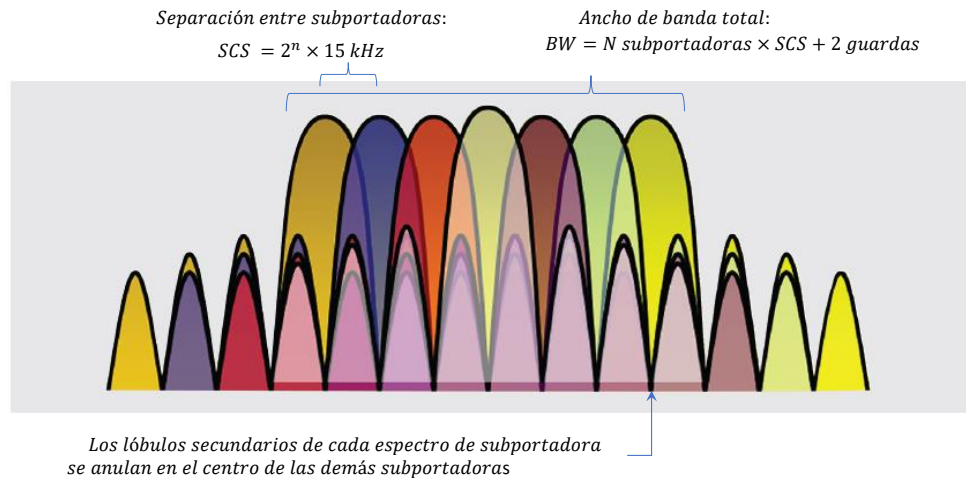


Figura 2.4: Construcción de OFDM a partir de su espectro. (Figura elaborada a partir de Keysight, Flexible OFDM Signal Generation, Analysis and Troubleshooting - Application Note.)

en frecuencia y se ecualizan con un valor numérico simple para representar la ganancia que requieren. Las subportadoras ocupan un ancho de banda fijo y no requieren espacio entre ellas, debido a que forman una base ortogonal. Es decir, que el valor de las componentes espectrales de cada una de ellas es cero en la frecuencia central de la siguiente, como se representa en la figura 2.4. Para lograrlo, la forma de pulso utilizada en cada una debe tener una Transformada de Fourier semejante a la función seno cardinal. Esta condición de ortogonalidad impone que la separación entre portadoras sea igual al inverso del tiempo de pulso.

Para conocer el estado del canal se deben enviar valores conocidos en algún momento y en algunas portadoras, y los valores medidos informan cuanto se debe amplificar cada una para recuperar el valor original. Interpolando entre frecuencias, e interpolando en el tiempo, entre esas señales conocidas se estima la ecualización completa del canal y su variación de forma ágil y sencilla. En LTE se hacía utilizando posiciones fijas que se repetían cíclicamente. En 5G, no hay señales de referencia fijas, se utilizan algunas señales conocidas, como las señales de sincronismo, y se completa esa información con señales de medida que son asignadas por la radio base para cada UE. Esto permite ahorrar energía y evitar usar señales que ocupan posiciones de posibles datos, cuando no se necesitan.

El ancho de banda total es consecuencia de multiplicar el espaciamiento por la cantidad de portadoras.

$$BW = SCS \times \text{Cantidad Total de Portadoras}$$

Por otro lado, la frecuencia de muestreo del OFDM sale de dividir la suma de las portadoras de la IFFT más el prefijo cíclico entre el tiempo de símbolo. La IFFT será del tamaño de la menor potencia de dos que sea mayor al número de subportadoras.

$$f_m = \frac{\text{muestras por simbolo}}{t_{sym}} = (\text{muestras por simbolo}) \times \Delta f_{SCS}$$

El oscilador del SDR en la maqueta deberá poder trabajar a una frecuencia múltiplo de la frecuencia de muestreo de la configuración a reproducir.

Del esquema clásico de OFDM presentado en la figura 2.5, todas las etapas de procesamiento digital están incluidas en la capa física baja (PHY low) de la pila de protocolos de NR y son hechas en la banda base de los gNB, o en la etapa de *Digital Unit* (DU), si la arquitectura del gNB es dividida. Las etapas recuadradas en azul, de conversión analógica digital, y multiplicación por oscilador local, son hechas en los *Radio Unit* (RU) de las radio bases comerciales tradicionales, como en LTE. En la maqueta de este proyecto, todas las etapas digitales se procesan en CPU x86 y las etapas de radio son SDR. La frecuencia de muestreo antes descrita será la base de tiempo de la transmisión de muestras complejas entre el conversor paralelo a serie y los conversores Digital a Analógico, e igual en el otro sentido. En el caso de los SDR del proyecto las muestras se transmiten encapsulada en IP, o por interfaces USB 3.0.

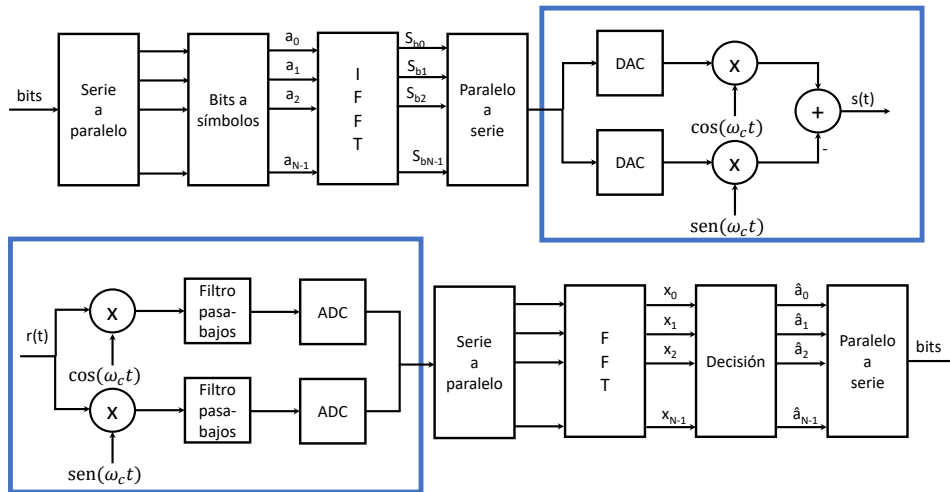


Figura 2.5: Esquema de bloques de OFDM. Se indican en azul las funciones hechas por los SDR [11].

Separación entre Portadoras

La numerología se refiere al conjunto de parámetros que definen la estructura de la capa física: el espaciado entre subportadoras (SCS), la duración del símbolo y la longitud del prefijo cíclico. Mientras que en LTE se utilizaba un SCS fijo de 15 kHz, con un tiempo de símbolo fijo. En NR puede tomar varios valores de la forma $SCS = 2^n \times 15 \text{ kHz}$, por lo que toma los valores de 15 kHz, 30 kHz, 60 kHz, 120 kHz y 240 kHz. Los tres primeros se utilizan en FR1 y los otros dos solo en en FR2, mientras que 60 kHz se usa en ambos rangos. Eso mantiene la ortogonalidad entre

2.4. Conceptos Claves para la Implementación

las subportadoras de la misma numerología y hace que las numerologías mayores no interfieran con las menores y minimicen la interferencia inter portadoras (ICI por su sigla en inglés) con las mayores, cuando hay diferente numerología en partes separadas del espectro. Esta relación permite también mantener la estructura de trama y los anchos de banda totales, cambiando el número de subportadoras. La numerología de los tipos NR se especifica en T.S. 38.211.

μ	$N_{\text{slot}}^{\text{slot}}$	$N_{\text{frame}, \mu}^{\text{frame}}$	$N_{\text{slot}}^{\text{subframe}, \mu}$
0	14	10	1
1	14	20	2
2	14	40	4
3	14	80	8
4	14	160	16

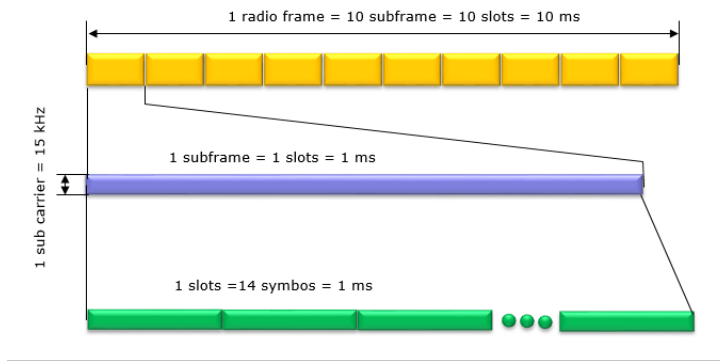


Figura 2.6: Estructura de trama, subtrama y slot SCS 15khz. [35]

μ	$N_{\text{slot}}^{\text{slot}}$	$N_{\text{frame}, \mu}^{\text{frame}}$	$N_{\text{slot}}^{\text{subframe}, \mu}$
0	14	10	1
1	14	20	2
2	14	40	4
3	14	80	8
4	14	160	16

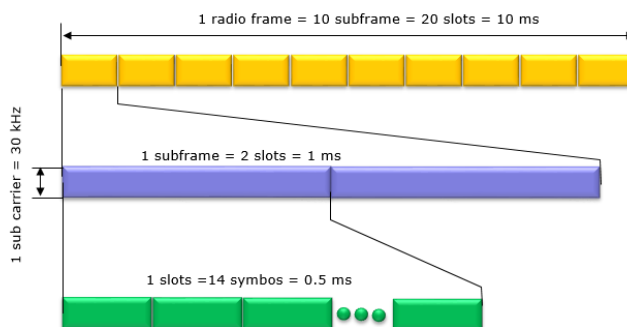


Figura 2.7: Estructura de trama, subtrama y slot SCS 30khz [35]

Capítulo 2. Redes Móviles 5G

Parameter / Numerlogy (u)	0	1	2	3	4
Subcarrier Spacing (Khz)	15	30	60	120	240
OFDM Symbol Duration (us)	66.67	33.33	16.67	8.33	4.17
Cyclic Prefix Duration (us)	4.69	2.34	1.17	0.57	0.29
OFDM Symbol including CP (us)	71.35	35.68	17.84	8.92	4.46

Figura 2.8: Duración de símbolo OFDM según la numerología.

Formato de Trama y de Slot

El flujo de información va organizado en tramas o *frames* que se repiten continuamente en períodos de 10 ms. Esta organización permite que un UE, una vez identificado el comienzo de una trama, pueda seguir interpretando correctamente todas las siguientes. Cuando llega al estado en que reconoce los inicios de los símbolos OFDM, los inicios de trama y se identifica la celda, se dice que está sincronizado. Este proceso se detalla más adelante.

Como se observa en las figuras 2.6 y 2.7 las tramas se subdividen en subtramas de 1 ms, y estas en slots. Cada slot tiene 14 símbolos OFDM. Como el largo del símbolo depende de la SCS, para cada separación habrá un diferente número de slots por subtrama. Cuando se duplica el SCS, el tiempo de símbolo se divide a la mitad, por lo que se duplica la cantidad de slots en la subtrama. La tabla de la figura 2.8 muestra los valores de los tiempos de símbolo (t_s) para cada SCS.

La mínima unidad de información a ser transmitida es el *resource element* (RE), que consiste en una portadora, durante un tiempo de símbolo. El número de subportadoras por símbolo OFDM depende del ancho de banda disponible y de la numerología utilizada. Como el máximo total posible es 275×12 [43], se obtienen anchos de banda de 50 MHz para SCS = 15 kHz, 100 MHz para SCS = 30 kHz, etc.

Los RE se pueden identificar en una estructura matricial de índices (k,t) que representan su posición en frecuencia y en tiempo dentro del slot. Como la cantidad de símbolos unitarios es demasiado alta, para hacer más eficiente su manejo, se agrupan en bloques de recursos físicos, PRB. Doce RE durante un símbolo forman un PRB. Al momento de asignar recursos, se adjudica una cantidad de PRB a lo largo de un período de tiempo. Lo habitual es que el elemento unitario en el tiempo sea un slot, pero en ocasiones, pueden hacerse asignaciones menores (minislot, de menos símbolos que un slot, para obtener latencias bajas).

El primer símbolo de cada slot se utiliza para transmitir el canal de control en DL, el PCCH, aunque dependiendo de la configuración pueden ser los dos o tres primeros. Allí se envía la información de la asignación de recursos en DL y UL (DCI), la ubicación de los pilotos requeridos para la demodulación (DMRS), la ubicación de las señales de medida y todo el seteo de control (Control REcourse SET, CORESET). Si se trabaja en TDD, algunos slots se usarán en DL y otros en UL. En TDD estático se define un ciclo de repetición en el cual se reitera la asignación. Entre el DL y el UL es necesario dejar una pequeña ventana de tiempo para evitar la superposición de señales débiles (UL) con señales fuertes (DL). Pero dejar un slot entero es demasiado, por eso se definen slot híbridos en los que algunos símbolos son de DL, otros de separación, y otros de UL. El formato de los slots

2.4. Conceptos Claves para la Implementación

híbridos indica cómo se utiliza cada uno de los símbolos dentro de ese slot.

Format	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	D	D	D	D	D	D	D	D	D	D	D	D	D	D
1	U	U	U	U	U	U	U	U	U	U	U	U	U	U
2	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3	D	D	D	D	D	D	D	D	D	D	D	D	D	F
4	D	D	D	D	D	D	D	D	D	D	D	D	F	F
5	D	D	D	D	D	D	D	D	D	D	D	F	F	F
6	D	D	D	D	D	D	D	D	D	D	F	F	F	F
7	D	D	D	D	D	D	D	D	D	F	F	F	F	F
8	F	F	F	F	F	F	F	F	F	F	F	F	F	U
9	F	F	F	F	F	F	F	F	F	F	F	F	F	U
10	F	U	U	U	U	U	U	U	U	U	U	U	U	U
11	F	F	U	U	U	U	U	U	U	U	U	U	U	U
12	F	F	F	U	U	U	U	U	U	U	U	U	U	U
13	F	F	F	F	U	U	U	U	U	U	U	U	U	U
14	F	F	F	F	F	U	U	U	U	U	U	U	U	U
15	F	F	F	F	F	F	U	U	U	U	U	U	U	U

Figura 2.9: Configuración de símbolos asignados a UL y DL dentro de un slot híbrido [3]. (figura de sharetechnote.com)

El número de combinaciones posibles de símbolos DL, símbolo UL y símbolo flexible dentro de una slot es muy alto, pero 3GPP solo permite 61 combinaciones de símbolos predefinidas dentro de un slot. En la tabla de la figura 2.9, que es parte de la tabla 11.1.1-1 de la norma TS 38.213 del 3GPP, se muestran quince como ejemplo.

2.4.3. Sincronización

Ocurren dos procesos de sincronización entre un UE y un gNB, uno en DL y otro en UL. La sincronización en DL ocurre cuando el UE está escuchando en la búsqueda de una celda. El UE busca identificar el comienzo exacto de una trama y el inicio de cada símbolo OFDM. El inicio del símbolo OFDM lo identifica mediante la correlación entre el prefijo cíclico y el final del símbolo. Para encontrar el inicio de trama debe identificar el *Sincronization Signal Block* (SSB), un conjunto de señales formado por la *Primary Sincronization Signal* (PSS), la *Secondary Sincronization Signal* (SSS) y el *Physical Broadcast Channel* (PBCH). Este bloque se retransmite periódicamente, típicamente cada 2 tramas (20 ms) pero puede variarse ese tiempo entre 5 ms y 160 ms. Y su ubicación en frecuencia, respecto del inicio de la banda o de la frecuencia central del canal, no está predeterminada.

El UE no conoce la separación entre subportadoras de la celda. Pero el SSB tiene SCS fija, depende solo de la banda y frecuencia donde esté ubicado el bloque. Entonces se dedica a buscar las señales de sincronismo, que son conocidas, a lo largo del tiempo y a lo ancho de las frecuencias. Una vez detectada la PSS y la SSS, recibe y decodifica la *Master Information Block* (MIB), que le informa sobre el SCS en uso, y sobre la ubicación del siguiente bloque de información SIB1 (*System Information Block Type 1*) que se transmite en el *Physical Downlink Shared Channel* (PDSCH). Puede haber hasta veintiún SIB diferentes. El SIB1

Capítulo 2. Redes Móviles 5G

proporciona información sobre la identidad de la celda y sobre la ubicación y período de reiteración de cada uno de los demás SIB en el PDSCH. Cada uno de estos agrega información sobre la trama y su estructura de control.

En resumen, cada teléfono que se inicia, debe buscar en las distintas bandas de frecuencias que maneja si hay celdas activas y lo hace recorriendo el espectro en la búsqueda de las señales de sincronismo. Una vez que las encuentra es capaz de recibir la información básica de la celda, y con ella comenzar a decodificar las tramas.

Una vez que tiene una celda de un gNB identificada y la está recibiendo puede iniciar el proceso de sincronización en UL, para poder transmitir. Un teléfono ya sincronizado solo puede transmitir en las frecuencias y ventanas de tiempo que el gNB le indica en los mensajes de control *Downlink Control Information* (DCI) y debe pedir agenda, si quiere empezar a transmitir nuevamente. Pero cuando el teléfono se inicia, o tras un período de desconexión, el gNB no sabe que está y por lo tanto no tiene agenda ni oportunidad de pedirla. Para resolver esa conexión inicial existe un espacio de tiempo y frecuencias donde se permite el intento de transmisión aleatoria, Conociendo la ubicación de ese canal de *Physical Random Acces Channel* (PRACH) se puede comenzar el proceso de acceso aleatorio (RACH). El UE transmite utilizando un número de identificación temporal para uso en ese canal autogenerado según una regla predefinida. En una secuencia de 4 mensajes con sus respuestas el UE se hace conocer por el gNB, obtiene un *Radio Network Temporary Identifier* (RNTI) para identificar en la celda sus asignaciones de *scheduling* y lo mantiene mientras esté conectado. Llegando al mensaje 4 de la secuencia de RACH, el UE está sincronizado en ambos sentidos.

En las redes inalámbricas los usuarios suelen estar largos períodos de tiempo sin necesidad de transmitir o recibir datos. Es conveniente que en esos períodos el dispositivo no tenga los receptores y transmisores encendidos, consumiendo energía y gastando baterías sin necesidad. En las redes móviles se definen diagramas de estados, y procedimientos para los cambios dependiendo de la actividad del tráfico. En NR hay tres estados definidos: RRC_CONNECTED, RRC_INACTIVE y RRC_IDLE, y su relación se muestra en la figura 2.10.

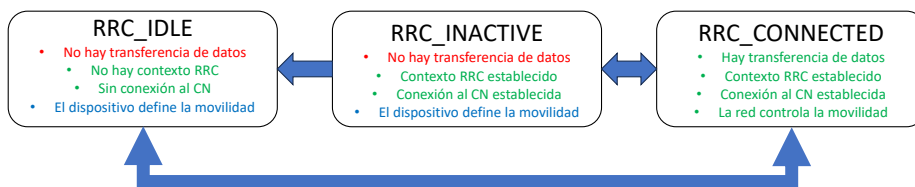


Figura 2.10: Diagrama de estados del Radio Resource Control. [13]

En el estado CONNECTED el dispositivo está ubicable por la RAN y el CN y la transmisión de datos es posible. En este estado quedan establecidos todos los parámetros necesarios para la comunicación, tanto para el dispositivo como para el CN. La movilidad es manejada por la red.

2.4. Conceptos Claves para la Implementación

En el estado IDLE no es posible transmitir datos, el dispositivo figura como desconectado de la red y solamente escucha en los tiempos de paging para saber si es llamado por la red. Para pasar a conectado es necesario repetir el proceso de RACH. El control de la movilidad lo tiene el dispositivo.

El estado INACTIVE es un estado nuevo que se introduce en NR y no existía en LTE original. El contexto se mantiene establecido, el bearer de radio por el que se maneja el canal de control sigue activo entre el CORE NETWORK y el gNB. No es posible la transferencia de datos y el no hay scheduling para este UE mientras el canal de control dedicado esté inactivo. Como para el CN el dispositivo está conectado, pasar a CONNECTED es muy rápido y no requiere el proceso de RACH ni el intercambio de información posterior con el CN. Esto es una ventaja importante tanto para la latencia como para el ahorro de energía respecto de los sistemas anteriores.

2.4.4. El tamaño del Bloque de Transporte (TBS)

El bloque de transporte en 5G NR es la carga útil que se pasa entre las capas MAC y PHY específicamente para el canal de datos compartidos PDSCH (*Physical Downlink Shared Channel*), PUSCH (*Physical Uplink Shared Channel*). Un bloque de transporte se somete al procesamiento de la capa PHY en el transmisor antes de ser mapeado en el PDSCH para su transmisión a través de la interfaz de radio.

Se agrega un bloque de transporte con CRC (*Cyclic Redundancy Code*) y se segmenta en bloques de código como se muestra a continuación. El bloque de transporte consta de hasta un millón de bits y el bloque de código consta de hasta 8448 bits. 2.11

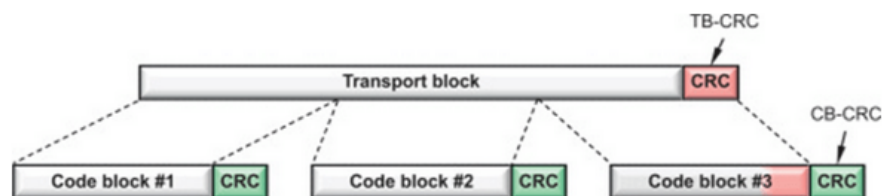


Figura 2.11: Diagrama de fragmentación del Bloque de Transporte en capa RLC. [41]

Factores que Afectan el TBS

El dispositivo que recibe datos en el PDSCH debe determinar el tamaño del bloque de transporte (TBS) antes de intentar decodificar los datos. El UE utiliza la información proporcionada por la señalización RRC y la información de control de enlace descendente (DCI) sobre el PDCCH dependiendo del tipo de programación, es decir, programación semiestática o dinámica.

En la figura 2.12 se muestra un diagrama simplificado para obtener el tamaño del bloque de transporte usando parámetros que afectan el tamaño del TB, que se

Capítulo 2. Redes Móviles 5G

obtienen del *Modulation Coding Scheme* (MCS) a partir de tablas de asignación normalizadas.

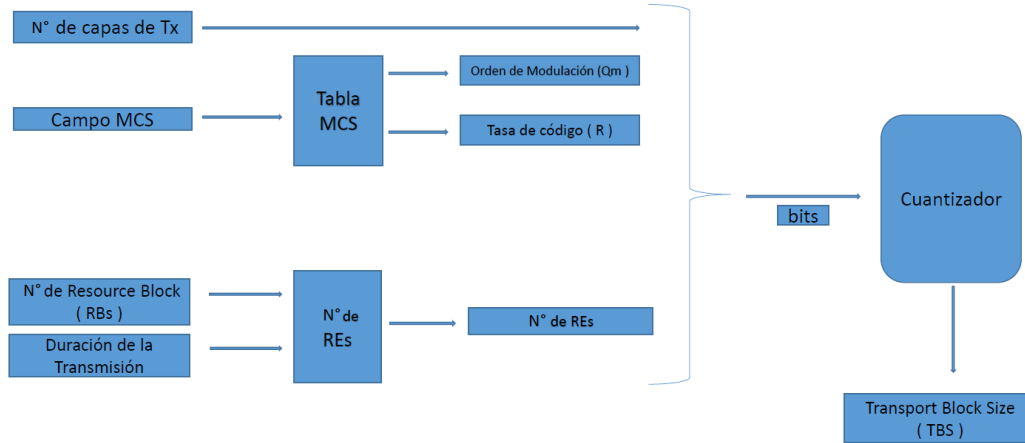


Figura 2.12: Diagrama simplificado para obtener el tamaño del bloque de transporte. Proporciona listas de parámetros que afectan el tamaño del TB.

Cómo Calcular el TBS

En 4G LTE, el tamaño del bloque de transporte se proporciona en una tabla que es una función del campo MCS y la asignación de bloques de recursos físicos (PRB). Sin embargo, 5G NR admite anchos de banda significativamente mayores, que varían de un UE a otro según la BWP en uso, junto con una amplia gama de duración del tiempo de transmisión a asignar, variaciones en la cantidad de encabezados de capa MAC, y una cantidad variable de señales para medir y mensajes de reportes CSI-RS. Esta gran variación de condiciones exigiría gran cantidad de tablas por opciones diferentes para determinar el tamaño de los bloques de transporte.

Dicho esquema se modifica siempre que cambie algunos de estos parámetros. Por lo tanto, 5G NR ha optado por un enfoque basado en fórmulas combinado con una tabla para los tamaños de bloques de transporte más pequeños para obtener la flexibilidad necesaria. En las siguientes ecuaciones se resume el cálculo del TBS.

Se evalúa N_{info} , el número total de bits de información, según la fórmula:

$$N_{info} = N_{re} \times R \times Q_m \times v \quad (2.1)$$

con

- N_{info} : número total de bits de información del TBS,
- N_{re} : número total de RE (*Resource Elements*) disponible para transferencia de datos,
- R : tasa de codificación,

2.4. Conceptos Claves para la Implementación

- Q_m : índice de modulación,
- v : número de capas.

Si se cumple la desigualdad siguiente, entonces el TBS se obtiene de la tabla 5.1.3.2-1 del 3GPP TS 38214.

$$N_{info} \leq 3824 \quad (2.2)$$

De lo contrario el TBS surge de la fórmula 2.1 cuantizada a valores múltiplos de potencias de dos, según un grupo de fórmulas que aplican a distintos casos dependientes del tamaño y de R , detallados en esa norma. El TBS debe ser cuantizado por potencias de dos iguales o mayores que 8, ya que se deben acomodar paquetes o fracciones de paquetes en Bytes.

2.4.5. HARQ Retransmisiones Compartidas entre Capa Dos y Capa Física

HARQ (Hybrid Automatic Repeat reQuest) es un mecanismo utilizado en redes de comunicaciones, incluyendo 5G, para mejorar la confiabilidad de la transmisión de datos. HARQ es una técnica de retransmisión que permite a un receptor solicitar la retransmisión de paquetes de datos que han sido recibidos de manera incorrecta o incompleta. La política de retransmisiones en 5G, que involucra el uso de HARQ, se diseña para maximizar la eficiencia de la transmisión y minimizar la pérdida de datos. A continuación, se describen aspectos clave de la política de retransmisiones en 5G:

HARQ en 5G

5G utiliza HARQ para garantizar que los datos se entreguen de manera confiable incluso en condiciones de señal débil o interferencia. Cuando un receptor detecta errores en los datos recibidos, solicita una retransmisión específica de los paquetes de datos perdidos o dañados en lugar de simplemente descartarlos.

Retransmisiones Incrementales

HARQ utiliza retransmisiones incrementales. En lugar de retransmitir todo el paquete de datos, el receptor solicita solo las porciones de datos que faltan o están dañadas. Esto ahorra ancho de banda y tiempo de transmisión.

Adaptación de Modulación y Codificación (AMC)

En 5G, HARQ puede estar integrado con técnicas de AMC. Esto significa que, si una transmisión inicial falla, el sistema puede intentar nuevamente con una modulación y codificación más robusta que sea menos susceptible a errores, lo que aumenta las posibilidades de una transmisión exitosa.

Capítulo 2. Redes Móviles 5G

Tiempo de Retransmisión

La política de retransmisiones en 5G tiene en cuenta el tiempo disponible para retransmitir datos. En aplicaciones de baja latencia se pueden establecer límites de tiempo estrictos para garantizar que las retransmisiones se realicen de manera oportuna.

Prioridades entre los Datos

En situaciones de congestión o alta demanda de red, es posible que se establezcan políticas de prioridades de datos. Esto significa que los datos críticos pueden recibir prioridad en las retransmisiones sobre los datos menos críticos.

Control de Errores

5G utiliza códigos de corrección de errores para detectar y corregir errores en las transmisiones. HARQ y estos códigos trabajan juntos para minimizar la probabilidad de pérdida de datos.

La política de retransmisiones en 5G se adapta dinámicamente a las condiciones de la red y al estado de la conexión para garantizar una transmisión confiable y eficiente de datos. Esto es especialmente importante en aplicaciones que requieren baja latencia y alta confiabilidad, como la transmisión de video en tiempo real, la comunicación entre vehículos autónomos y las aplicaciones de misión crítica.

A continuación, en la figura 2.13 se ilustran algunos ejemplos típicos de funcionamiento de HARQ.

Caso 1: Aprobado en la Primera Recepción

Transmisión: El gNB (estación base) transmite PDSCH con rv 0 al UE (equipo de usuario) a través del PDSCH.

Recepción: El UE recibe los datos y realiza la detección de errores utilizando CRC (Cyclic Redundancy Check).

ACK/NACK: Si el CRC está bien, el UE envía un mensaje ACK al gNB a través de PUCCH o PUSCH, lo que indica una recepción exitosa. No hay necesidad de retransmisiones en este caso. No es necesario guardar los datos en la memoria intermedia HARQ y el UE reenvía los datos decodificados a una capa superior.

Caso 2 : Pase con Retransmisión

Primera transmisión: el gNB transmite PDSCH con rv 0 (redundancy version) al UE a través del PDSCH. **Primera recepción:** el UE recibe los datos y realiza la detección de errores utilizando el CRC. El CRC falla, lo que indica un error de decodificación.

Primer ACK/NACK: el UE envía un mensaje NACK al gNB a través del PUCCH o PUSCH, indicando un error de decodificación.

2.4. Conceptos Claves para la Implementación

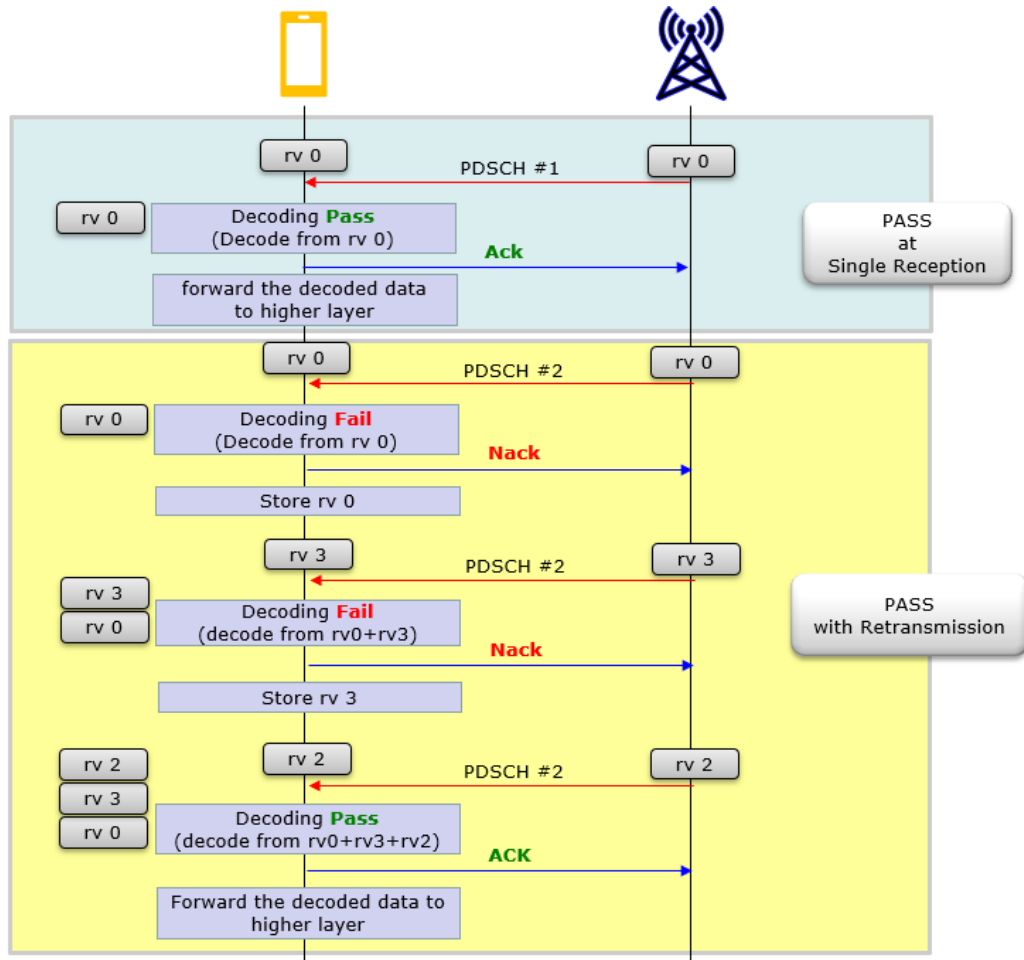


Figura 2.13: Diagrama de funcionamiento de HARQ. [34]

Primera retransmisión: según el NACK, el gNB retransmite los datos con un RV diferente (por ejemplo, RV=3) para aumentar la probabilidad de una decodificación exitosa.

Segunda recepción: el UE recibe los datos retransmitidos y realiza la detección de errores utilizando el CRC. El CRC vuelve a fallar, lo que indica otro error de decodificación. Cuando el UE decodifica datos, combina los datos anteriores almacenados en el búfer y los datos actuales que acaba de recibir, y luego decodifica los datos combinados.

Segundo ACK/NACK: el UE envía otro mensaje NACK al eNodeB, indicando un segundo fallo de decodificación.

Segunda retransmisión: según el segundo NACK, el gNB retransmite los datos nuevamente con un RV diferente (por ejemplo, RV=2) para aumentar aún más la probabilidad de una decodificación exitosa.

Tercera recepción: el UE recibe los datos retransmitidos y realiza la detección de errores utilizando el CRC. El CRC está bien esta vez, lo que indica una decodificación exitosa. Cuando el UE decodifica datos, combina todos los datos

anteriores almacenados en el búfer y los datos actuales que acaba de recibir, y luego decodifica los datos combinados. Tercer ACK/NACK: el UE envía un mensaje ACK al gNB, indicando una recepción exitosa después del tercer intento de transmisión.

2.5. Evolución y Desafíos

5G es una tecnología en evolución. A su definición inicial en la *Release 15*, se agregaron actualizaciones y mejoras en la 16 y la 17. La *Release 18* está en discusión y la 19 está planificada. En cada versión se incluyen mejoras sobre las funcionalidades existentes y se agregan funcionalidades nuevas [30]. En la *Release 16* las mejoras más notables son:

- ***Multiple-input, multiple-output y Beamforming***, se agregan mensajes CSI de retorno y como soporte para transmisiones desde múltiples puntos a un único UE,
- ***Dynamic Spectrum Sharing***, un método para compartir la misma frecuencia entre LTE y NR en escenarios de transición,
- ***Dual Connectivity and Carrier Aggregation***, se reducen los tiempos de seteo para estas configuraciones,
- **Ahorro de energía en UE**, se agrega una señal de control para despertar un UE, mejorando el consumo a partir de la interacción con el *scheduler*.

Además, entre las funciones nuevas se destacan:

- ***Integrated access and backhaul (IAB)***, una función que permite que un gNB con buena transmisión pueda oficiar de donante, y darle servicio de transmisión a otro gNB a través de la propia interfaz NR,
- **NR en espectro no licenciado**, definiciones de esquemas de conexión y métodos de acceso al medio para utilizar las bandas no licenciadas en forma compartida con las demás aplicaciones usuarias de estas bandas,
- **Posicionamiento**, función que permite determinar la posición del móvil a partir de medidas de tiempo de ida y vuelta (RTT, por round-trip time) a distintas radio bases. Se definen señales de medida específicas llamadas *Positioning Reference Signal* (PRS) que permiten buena precisión y complementan a los servicios satelitales con buenos resultados en ambientes interiores.

En la *Release 17* además de mejoras en múltiples áreas se destaca:

- **Espectro de 52.6 GHz a 71 GHz** agregado a FR2, incluyendo la banda no licenciada de 60 GHz,
- **RedCap y NR-Light** Nuevos dispositivos de capacidades reducidas, con valores intermedios entre los de NB-IOT y un equipo NR completo,

2.5. Evolución y Desafíos

- **NTN** Uso de redes no terrestres como cobertura alternativa para móviles en áreas alejadas. Enfocadas en el uso de sistemas de satélites geoestacionarios, satélites de órbitas de baja altura (LEO) y globos o sistemas en la alta atmósfera.
- **NR Sidelink** Conexión directa entre móviles.

La futura 6G

En el horizonte de la evolución de 5G el bosquejo de la futura red 6G ya está sobre la mesa. Además de dar servicios para un nivel superior de conducción automatizada y de fabricación inteligente, 6G implementará aplicaciones innovadoras al combinar imágenes, sensores y sincronización precisa de la movilidad, que harán un verdadero aprovechamiento de la inteligencia artificial (AI) y de las redes inteligentes.

Para cumplir con este cometido, se requiere diseñar las redes 6G para que mejoren sus *Key Performance Indicator* (KPI), de manera de aumentar sus prestaciones entre 10 y 100 veces con respecto a 5G. Los primeros objetivos apuntan velocidades de datos máximas, latencia y densidad, pero 6G también le da relevancia a la mejora de los KPI de jitter. En la figura 2.14 se resumen los principales objetivos de la tecnología.

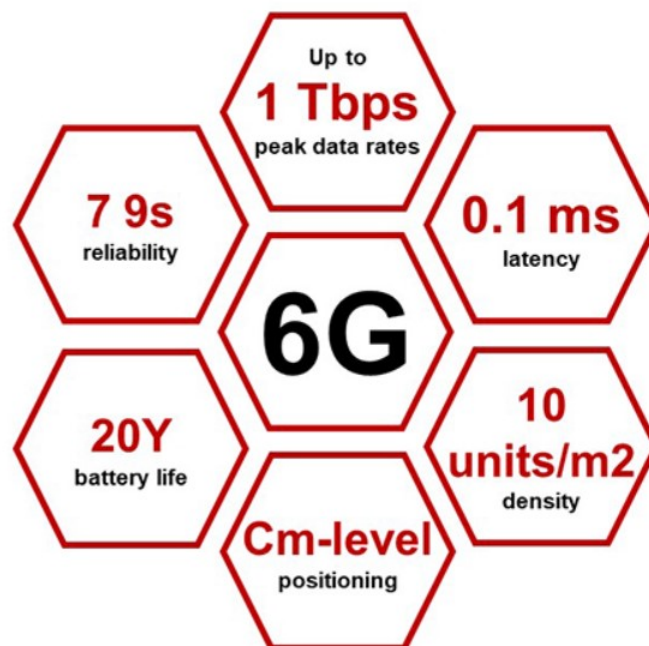


Figura 2.14: Los paradigmas de 6G, implica la incorporaciones de frecuencias de THz y de AI. [12]

La necesidad de ampliar los anchos de banda requiere el uso de frecuencias superiores a los 100 GHz (sub-Tera Hertz) implementadas en redes de corto alcance.

Capítulo 2. Redes Móviles 5G

Para lograr la sincronización precisa es necesario rediseñar las redes, incorporar nueva electrónica, optimizar la forma en que se operan y la incorporación de más recursos informáticos. La mejora en los aspectos de seguridad es clave, prevenir ataques y ser capaz de recuperarse de desastres, haciendo énfasis en la privacidad de los usuarios.

Otro aspecto clave es la posible incorporación de frecuencias del orden de Tera Hertz (THz), esto implica longitudes de ondas aún más cortas. Por lo tanto las pérdidas en la propagación atmosférica son aún mayores, con anchos de haz también menores, esto implica una mayor integración antenas, dispositivos y desarrollos de software para los conceptos de *beamforming*. En los objetivos aspiracionales también hay aspectos vinculados al uso de AI en todos los niveles de control y operación, en la optimización de la banda base y técnicas de radio frecuencia, y hay objetivos más exigentes respecto de los menores consumos de energía de los dispositivos.

Open RAN

A esta evolución guiada por el 3GPP se le suma el desafío del movimiento OPEN RAN. Desde la O-RAN Alliance y otras instituciones se impulsa la construcción de redes de acceso abiertas e inteligentes en todas las generaciones. Esto se traduce en la normalización de las funciones e interfaces internas de los equipos, para hacerlos compatibles entre proveedores. Este movimiento agrega herramientas e interfaces con aportes sustantivos en la inclusión de herramientas de AI/ML para la gestión de la red, y en la definición de arquitecturas y casos de uso. Desde 2010, con la aparición de 4G, un grupo de operadores viene manifestando su interés de moverse de redes construidas con hardware específico a redes de software especializado sobre hardware multipropósito y entornos virtualizados. Ese camino está muy avanzado en la red de core. La Open-RAN Alliance (O-RAN) está trabajando para lograrlo también en la Radio Access Network.

Este grupo se dedica a especificar los elementos constitutivos de una radiobase, y las interfaces entre ellos, con un nivel de detalle suficiente para que dos partes independientes, que cumplan con su normativa, interactúen correctamente garantizando la funcionalidad del sistema. Eso permite correr muchas de las funciones en hardware genérico o virtualizado y construir efectivamente redes multivendor. Impulsa el surgimiento de nuevos proveedores enfocados en elementos de su especialización. Como contraparte, la integración y validación del sistema es un desafío para los operadores.

Un segundo aporte importante de O-RAN es la definición de funciones de control inteligente a nivel de la RAN que constituyen un entorno de trabajo para la aplicación de herramientas de AI/ML en la gestión de la red. O-RAN no es solo 5G, pero 5G le ha dado un impulso grande a O-RAN dada la convergencia de intereses. Es relevante para la proyección futura de la maqueta resultante de este proyecto su posible evolución incluyendo estas opciones. Las dos alternativas estudiadas, y en particular OAI, la elegida, son parte de la O-RAN Alliance, y trabajan en desarrollos que incluyan sus opciones.

Investigación

Un área central en la investigación académica y en la implementación práctica por parte de los operadores actualmente se enfoca en las estrategias para implementar servicios diferenciados utilizando *Network Slicing*. Es un desafío diseñar la división de los tipos de servicios en *slices* de red diferentes, que atraviesan todos los elementos y capas de la red, con mecanismos de manejo de recursos diferentes, según objetivos diferentes para cada tipo de servicio ofrecido. Calidades de servicio diferentes en tramos de transmisión, instancias de network funcion con políticas particulares en el core, y manejo de espaciamiento de portadoras, bandwidth parts, o selección de portadoras, son elementos que permiten comportamientos optimizados hacia distintos objetivos según los objetivos de servicio considerados. Esta dispersión entre métodos diferentes solo se puede implementar gracias a la gran flexibilidad del diseño base de 5G. Pero también imponen una serie de complicaciones a la hora de optimizar el uso de recursos, ya que no solo se deben considerar las prioridades entre UE, sino que también las prioridades entre *slices*, y a su vez entre los UE de ese *slice*, con recursos de radio de nodos diferente, pero compitiendo por recursos de transmisión o de CN, lo que permite imaginar fácilmente situaciones de antagonismo entre intereses. El *Network Slicing* y su manejo de los recursos, las estrategias de división y virtualización de la RAN, y su interacción con los sistemas de transporte y de distribución de sincronismo son áreas temáticas que están lejos de tener una solución cerrada.

No son las únicas áreas de investigación. Todas las mejoras mencionadas antes, están en desarrollo permanente y requieren investigación para su evaluación y optimización. En paralelo con las evoluciones técnicas, están los desarrollos de campos de aplicación. Las generaciones previas se desarrollaron sobre la base de un servicio masivo y uniforme y se financiaron con el crecimiento continuo del mercado. Hoy, el acceso a internet móvil de banda ancha es cuasi universal con más teléfonos que habitantes, lo que acota el crecimiento al incremento del tráfico en un universo de clientes estable. La financiación del desarrollo de nuevas tecnologías está condicionado a resolver nuevos problemas, o a resolver mucho mejor, con impacto económico, los problemas anteriores. Lo visto hasta ahora mejora la eficiencia espectral y la capacidad de transferencia, pero no es claro que en todos los casos sea suficiente para valorizar la nueva tecnología y repagar la inversión que demanda. La exploración de nichos de negocios diferentes, en mercados verticales y de aplicaciones con mayor valor agregado, pueden permitir pagar parte de ese desarrollo. Es así que un foco importante de la investigación está concentrada en casos de uso particulares.

Esta variedad de escenarios impulsa muchos campos de investigación académica teórica y aplicada y requiere también de la formación de profesionales para su desarrollo, despliegue y utilización. Esto abre las puertas para un sinnúmero de innovaciones particulares, que necesitan probarse y validarse en laboratorios adecuados primero y con redes reales después. En todos estos casos, maquetas como la de este trabajo tienen una función que cumplir.

2.6. Resumen

En este capítulo, se repasaron los fundamentos y desafíos de la tecnología 5G, describiendo sus objetivos y características, así como analizando algunos elementos que la definen. También se describieron algunos conceptos técnicos clave para la implementación de la maqueta.

Con estas herramientas, se abordará la selección de la plataforma de software para la implementación de la maqueta de red, objeto del proyecto. Se evaluarán dos opciones: OpenAirInterface y srsRAN, buscando la más útil tanto para la implementación actual como para considerar su uso y desarrollo futuro de la herramienta.

Adicionalmente, en este capítulo también se dedicó un párrafo a discutir las perspectivas y avances en la futura tecnología 6G, anticipando los posibles cambios y desafíos que esta próxima generación de tecnología móvil traerá consigo.

Capítulo 3

Selección de Plataforma de Software

El primer desafío es identificar cuál es la plataforma adecuada para implementar la Maqueta de 5G. Son requisitos básicos para los objetivos del proyecto que sea de código abierto y de acceso gratuito. También debe ser capaz de soportar la implementación de una red de 5G *Stand Alone* (SA) con interfaces de radio activas, es decir, el desarrollo de la *Radio Access Network* (RAN) debe ser lo suficientemente completo como para ser factible de implementar.

Este capítulo estudia las plataformas existentes, las compara y prueba y fundamenta la decisión tomada. Fue escrito en enero de 2023 al momento de definir la instalación. Se optó por mantenerlo sin cambios al cierre del proyecto. Eso explica las versiones, facilidades y *Roadmaps* de hace un año. Las actualizaciones en los sistemas evaluados son continuas, de modo que para conocer el estado actual de los desarrollos es mejor dirigirse a las páginas web de las comunidades de desarrollo.

3.1. Las Plataformas Evaluadas

Los candidatos iniciales son OpenAirInterfaceTM (OAI) [24] y Software Radio System Radio Access Network (srsRAN) [39], dos propuestas que a priori parecen ser las mejores disponibles. OAI es un desarrollo impulsado por la OpenAirInterface Software Alliance, una asociación sin fines de lucro de muchas empresas e instituciones que impulsan una comunidad de desarrollo. Es de código abierto y hecho mayormente en C/C++. Su distribución es gratuita para propósitos académicos o de investigación no comercial, bajo una licencia Apache v2.0 modificada. Este sistema está organizado en cuatro proyectos [8], OAI-RAN5G, Core Network (5G-CN), OAI MOSAIC5G y Continuous Integration / Continuous Deployment (CI/CD). OAI RAN5G distribuye la implementación de un enhanced NodeB-Long Term Evolution (eNB-LTE) y un new generation NodeB New Radio (gNB-NR) más un User Equipment LTE (LTE_UE) y un User Equipment New Radio para 5G (NR_UE). Los dispositivos de software corren sobre hardware comercial (COTS Commercial-off-the-shelf) con sistemas operativos Linux y utilizan Software Defined Radio (SDR) de varios fabricantes. En particular se utilizan los Universal Software Radio Peripheral (USRP) de Ettus [33], que se utilizan en este proyecto.

Capítulo 3. Selección de Plataforma de Software

5G CN desarrolla el Core Network de 5G. MOSAIC5G implementa un orquestador, un controlador y las interfaces necesarias para implementar la red virtualizada, y agregando controladores inteligentes a la RAN (RIC) como entorno de trabajo para el uso de herramientas de inteligencia artificial en la gestión de la red. Utiliza las aplicaciones de OAI como infraestructura. Su utilización queda fuera del alcance de este proyecto. Por último, el proyecto CI/CD es el responsable de mantener la calidad y compatibilidad de cada *merge* de los distintos proyectos individualmente y entre si mediante auditorías y ensayos.

srsRAN es una implementación en software de un sistema 4G y 5G, que incluye una aplicación eNB/gNB, un UE y un Evolved Packet Network (EPC) de 4G. Es compatible con CN 5G de terceros, como veremos más adelante. Es desarrollado por la empresa Software Radio System y por una comunidad de usuarios por ellos impulsada. Está escrito en C/C++ y es distribuido bajo una licencia AGPL versión 3. Es compatible con ambientes Ubuntu y Fedora sobre hardware comercial. Funciona con los principales SDR del mercado e incluye un simulador del canal de radio para pruebas sin interfaces de radio real.

Una primera búsqueda identifica varios desarrollos alternativos. Algunos son de tipo comercial, como Amarisoft que tiene una maqueta propietaria AMARI Callbox [9] con un desarrollo avanzado. También NVIDIA con su framework aerial [27], que es propietario pero se cede su uso con fines académicos, limitándose al testeo de interfaces entre sistemas de software con hardware dedicado o de aceleración. Otros son desarrollos parciales, de simuladores o de partes del sistema, como Open 5GCs [28] o free5GC [17] que se limitan a core network. [22] Algunos desarrollos que fueron populares quedaron por el camino, como OpenLTE [29] que dejó de evolucionar hace dos años sin dar el salto a 5G. Otros se fusionaron, como FlexRAN [26] que se integró en el proyecto Mosaic5G de OAI.

La búsqueda confirma que las dos plataformas con un desarrollo avanzado de RAN 5G en código abierto son las mencionadas OAI y srsRAN. Ambas soportan la implementación de una red 5G NSA y 5G SA. Las dos tienen publicado el software para instalar un gNB y un UE con esas facilidades. OAI también tiene implementado un Core Network (CN). srsRAN no lo tiene, pero tiene publicados ejemplos de instalación con 5GCs CN y fue el utilizado en las pruebas con srsRAN para este trabajo. Las dos plataformas incluyen una herramienta de conexión de red entre gNB y UE emulando el canal de radio y tienen logs e interfaces inspeccionables con wireshark u otro sniffer para analizar paquetes en los distintos protocolos implementados. Ante esta situación se limita el estudio a estas dos plataformas.

Se divide el resto del capítulo en el estudio de tres aspectos, la factibilidad de alcanzar el objetivo con la plataforma, el estado de avance del desarrollo de cada una, y la institucionalidad y actividad empresarial y académica vinculada a las plataformas OAI y srsRAN. El resumen del análisis se presenta al final del capítulo, justificando la elección de la plataforma a utilizar en este proyecto.

3.2. Factibilidad

Las dos plataformas son evoluciones del software de código abierto para implementar redes LTE de ambas comunidades. La existencia de esos desarrollos previos, robustos y bien probados, le dan credibilidad a la evolución actual. En un trabajo de evaluación comparativa de ambos desarrollos previos, hecho en 2017, se muestra que los dos sistemas LTE tenían una evolución y una performance comparable [18]. Las pruebas se hicieron sobre el mismo hardware de PC y usando SDR Ettus B210, con la misma CN OAI para ambos eNB y se conectaron como UE un teléfono LG Nexus 5, un dongle USB Huawei e3272-153 y el UE de srsLTE con un USRP B210, a las dos plataformas. El *throughput* y la latencia fueron del mismo orden en todos los casos. En la comparación de tiempos de establecimiento del servicio al iniciar la red y de tiempos de conexión de un UE, también fueron del mismo orden, con una tendencia a ser menos dispersos los resultados con srsRAN. En cambio en el uso de los recursos de hardware de las PC OAI fue mucho más eficiente. Se resalta el hecho de que se espera una demanda de recursos mucho mayor en redes 5G que en la red 4G con que se hizo esa comparación. Finalmente, en opinión de esos autores la estructura del software y su presentación es más simple y accesible para su comprensión y modificación en srsRAN que en OAI.

Es claro que esta comparación no se puede extender a las versiones actuales, con otra tecnología, pero resulta un antecedente fundamental para asegurar que la estructura de base de ambos desarrollos es suficientemente buena como para considerar viable su utilización. Trabajos en distintas universidades muestran casos de aplicación exitosa de ambas plataformas, mencionando a modo de ejemplo el prototipado de una maqueta para O-RAN con srsRAN [44] o el despliegue de una red 5G NSA con OAI [32].

3.3. Avance de los Desarrollos

Ambos desarrollos están en evolución, por lo que la presente evaluación resulta de revisar la última información publicada y revisada para este trabajo a noviembre de 2022. Seguramente tengan cambios incrementales en el corto plazo, y se pudo omitir involuntariamente otros datos recientes, debido al intenso trabajo en el área.

3.3.1. srsRAN

La versión actual es la 22.10 de noviembre de 2022. Se anuncia una versión nueva con cambios de arquitectura del gNB para enero de 2023, pero no se publican *roadmaps* detallados con la evolución planificada. Según la web de srsRAN las principales características soportadas por esta versión son:

1. Separaciones entre subportadoras: 15 kHz.
2. Bandas soportadas: FR1.
3. Anchos de banda: 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz y 20 MHz.

Capítulo 3. Selección de Plataforma de Software

4. Duplexado: FDD en gNB. FDD + TDD en UE.
5. Carrier Agregation: Sí, una segunda portadora.
6. Performance: DL+Carrier Agregation 150 Mbps.
7. Performance: DL SISO 20 MHz 75 Mbps.
8. Performance: UL 20 MHz 50 Mbps con UE COTS.
9. Máximo BW admitido por el UE en SA: 10MHz
10. Soporte de movilidad en 5G NSA y 5G SA: sí.
11. interfaces OPEN RAN no implementadas.

3.3.2. OpenAirInterface

Desde 2020 OAI dejó de publicar versiones cerradas, siendo la última la 1.22 y corresponde al software presente en el *branch* “*master*” del GitLab *openairinterface5G*, que tiene como última publicación agosto de 2020 y corresponde a la última versión solo LTE. Las versiones que incluyen 5G están todas en el *branch* “*develop*” y se identifican por la semana de la interacción. La última revisada al momento de hacer la evaluación es la “*integration_2022_wk51*”. El detalle de las características a este momento fue presentado en el Fall 2022 OpenAirInterface North American Workshop [15] y se resume a continuación.

1. Separaciones entre subportadoras: 15 kHz, 30 kHz (en FR1) y 120 kHz (en FR2).
2. Bandas soportadas: FR1 y FR2 implementada con Frecuencia Intermedia de 5GHz, por la no disponibilidad de SDR en FR2.
3. Anchos de banda: 20/40/50/60/80/100 MHz (FR1), 50/100MHz (FR2)
4. Duplexado: FDD y TDD estático y TDD dinámico.
5. Carrier Agregation: No.
6. MIMO DL: 2x2.
7. Soporta modulaciones de 256QAM DL
8. Performance: DL 2 Layers, 100MHz 820 Mbps.
9. Performance: DL SISO 1 Layer 60 MHz 201 Mbps.
10. Performance: UL 40 MHz 1 layer 40 Mbps.
11. interfaces OPEN RAN implementadas parcialmente: F1-C and F1-U, E2, 5G FAPI, Fronthaul. En roadmap está el mejorarlas y agregar E1 y O1 en 2023.

Además de mostrar un avance mucho mayor en los desarrollos, OAI hace públicos sus *Roadmaps* en la misma presentación. Se muestran en las figuras 3.1 y 3.2.

3.4. Instituciones, Foros y Actividades.

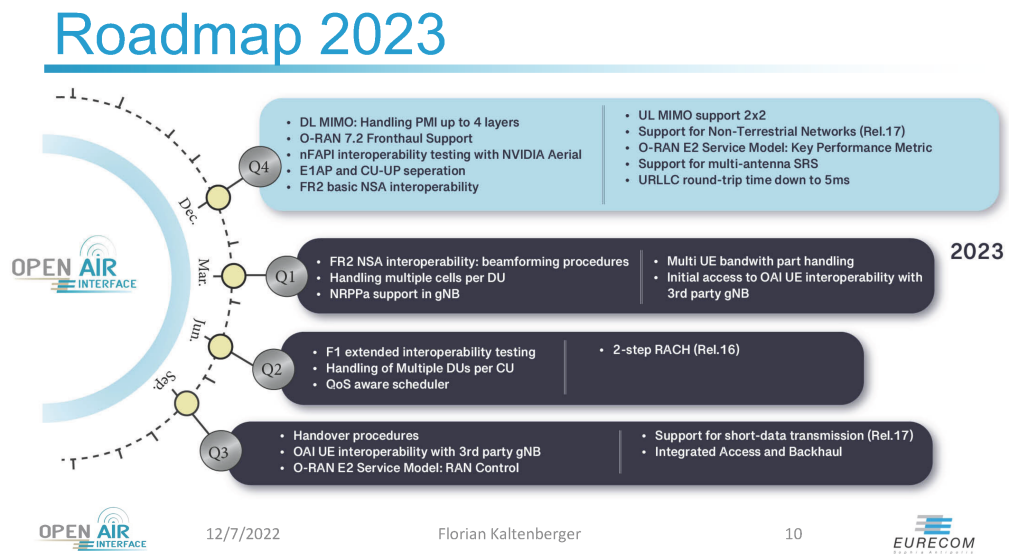


Figura 3.1: RAN Roadmap 2022/2023 presentado en el *Fall 2022 OpenAirInterface North American Workshop*. Presenta las perspectivas de desarrollo en el momento en que se tomó la decisión.

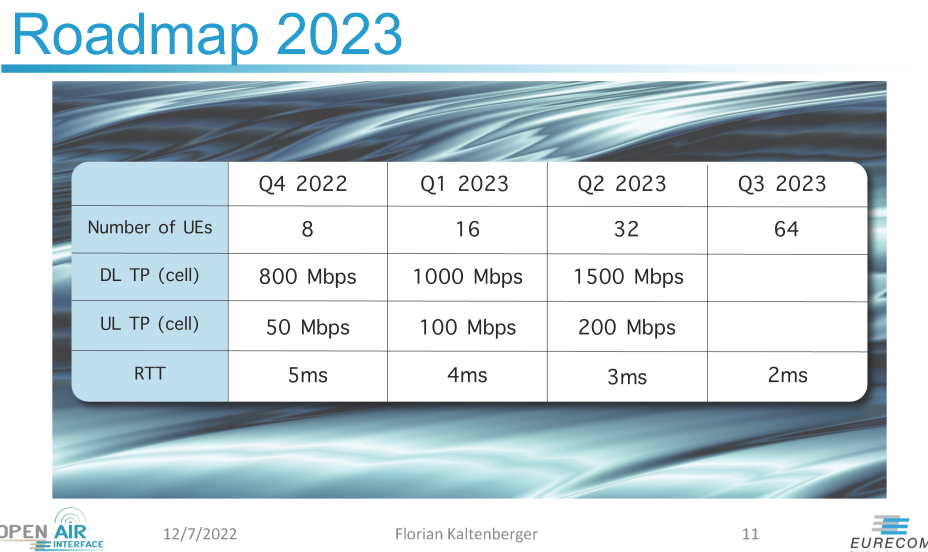


Figura 3.2: RAN Roadmap 2022/2023, continuación.

3.4. Instituciones, Foros y Actividades.

La razón de esta sección es estimar la dimensión y la calidad de las comunidades detrás de ambos desarrollos, para proyectar su capacidad de evolución y también sus riesgos.

La comunidad de srsRAN comenzó como un impulso individual de académicos

Capítulo 3. Selección de Plataforma de Software

independientes, que luego formaron la empresa Software Radio Systems (SRS) de base en Irlanda, que comercializa soluciones basadas en el software srsRAN y servicios asociados. Esta condición es a la vez su mayor fortaleza y su principal riesgo. El éxito de la empresa depende del avance continuo del desarrollo, lo que garantiza que esté en el foco de su atención. Pero un fracaso técnico, comercial o financiero, pone en riesgo a la empresa y a la vez al proyecto. Como su equipo de desarrollo principal es acotado se mantiene ordenado el software y actualizada la documentación. A la vez, eso hace que se mantenga una política de liberar solo dos versiones al año, enlenteciendo el desarrollo. De hecho no han habido nuevas funcionalidades en los seis primeros meses de este proyecto.

El proyecto OAI fue fundado por Eurecom, un instituto que imparte enseñanza de grado y posgrado y realiza investigación en comunicaciones y tecnologías de la información. Hoy es manejado por OpenAir Software Alliance, una organización sin fines de lucro dirigida y apoyada por múltiples empresas. Mientras que su dirección ejecutiva está integrada por profesores de Eurocom, su dirección estratégica está integrada por representantes de Meta, Nokia, Fujitsu, Qualcomm, National Instrument, Xilinx Inc., PAWR Project Office, Orange y Vodafone, entre otros. Un gran número de empresas, universidades y organizaciones se declaran miembros de la alianza [16]. Y otras, como AT&T, Ericsson o The Linux Foundation, participan con ponencias en sus eventos, aún sin integrarse formalmente.

El git del proyecto tiene dos *branches* públicos principales, el *master branch* y el *develop branch*. El *master* mantiene la versión 1.22, última del sistema solo LTE, y el *develop*, donde se trabaja el desarrollo de 5G. Se libera una nueva compilación cada una o dos semanas. Hay múltiples desarrollos simultáneos y accesibles, con decenas de desarrolladores muy activos. Las correcciones, mejoras y nuevas funciones se implementan continuamente, lo que dificulta la actualización de la documentación.

3.5. Instalaciones Preliminares de Prueba

Se realizaron instalaciones parciales de los sistemas en evaluación, para contar con elementos de decisión propios. Se utilizaron computadores propios, con características inferiores a las sugeridas. Se realizó la instalación completa de un CN de 5GCs, un gNB y un UE de srsRAN en máquinas virtuales, y se pudo establecer una conexión exitosa desde el UE, utilizando la conexión virtual ZeroMQ provista por este sistema para no usar SDR. En el Apéndice B se documenta dicha instalación y las pruebas realizadas. Fue posible hacer la puesta en marcha, siguiendo los tutoriales oficiales de srsRAN y resolviendo algunos problemas puntuales del sistema implementado.

Del mismo modo se realizó una primera instalación de OAI. Este sistema desaconseja hacer instalaciones en máquinas virtuales, por lo que se instalaron todas las componentes del sistema en una misma laptop. En esas condiciones se estableció una conexión entre UE y gNB utilizando el simulador de RF, que se muestra en la figura 3.3. Este simulador se explica en el Capítulo 4.4

```

[NR_MAC]   Frame.Slot 384.0
UE RNTI 168d (1) PH 0 dB PCMAX 0 dBm, average RSRP -44 (16 meas)
UE 168d: dlsch_rounnds 732/2/0/0, dlsch_errors 0, pucch0_DTX 2, BLER 0.00000 MCS 9
UE 168d: dlsch_total_bytes 90193
UE 168d: ulsch_rounnds 378/0/0/0, ulsch_DTX 0, ulsch_errors 0, BLER 0.00000 MCS 9
UE 168d: ulsch_total_bytes_scheduled 43848, ulsch_total_bytes_received 43848
UE 168d: LCID 1: 631 bytes TX
UE 168d: LCID 4: 12 bytes TX
UE 168d: LCID 4: 216 bytes RX

[NR_MAC]   Frame.Slot 512.0
UE RNTI 168d (1) PH 0 dB PCMAX 0 dBm, average RSRP -44 (16 meas)
UE 168d: dlsch_rounnds 745/2/0/0, dlsch_errors 0, pucch0_DTX 2, BLER 0.00000 MCS 9
UE 168d: dlsch_total_bytes 91792
UE 168d: ulsch_rounnds 385/0/0/0, ulsch_DTX 0, ulsch_errors 0, BLER 0.00000 MCS 9
UE 168d: ulsch_total_bytes_scheduled 44660, ulsch_total_bytes_received 44660
UE 168d: LCID 1: 631 bytes TX
UE 168d: LCID 4: 15 bytes TX
UE 168d: LCID 4: 270 bytes RX

```

(a) Consola del gNB mostrando el estado de enlace de un UE.

```

[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 465/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 471/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 478/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 484/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 490/2/0
[NR_PHY] =====
[NR_PHY] =====

```

(b) Consola del UE mostrando la cantidad de *PID* recibidos y las retransmisiones en primera y segunda instancia.

Figura 3.3: Conexión de prueba sin radios entre un UE y un gNB en una instalación preliminar de la plataforma OAI.

El detalle de esta instalación no tiene diferencias relevantes con la instalación completa con SDR, que se describe en el capítulo siguiente.

3.6. Selección

En la Tabla 3.1 se muestra que el avance de la implementación de 5G es mucho mayor en OAI. En el estudio previo se vió que la variedad de implementaciones y áreas de estudio en sus comunidades es más amplia e interesante en OAI y que en función de su amplia aceptación en la industria parece encaminada a convertirse en

Capítulo 3. Selección de Plataforma de Software

	srsRAN	OAI
Generalidades		
Versión actual	22.04.01	2022.wk51
UE 5G SA y NSA	Sí. 10MHz BW	Sí. Sin certificar otros gNB.
gNB SA y NSA	Sí	Sí
CN	No. Usa Open5GS.	Sí
RAN		
Subcarrier spacings:	15 kHz	15 kHz, 30 kHz, 60 kHz, 120 kHz
Bandwidth:	1.4/3/5/10/15/20 MHz	10/20/40/60/80/100 MHz (FR1) 50/100 MHz (FR2)
Duplexing:	FDD	FDD, TDD, dynamic TDD

Tabla 3.1: Tabla comparativa de características técnicas entre srsRAN y OAI a la fecha de la selección.

un estándar de facto. También se reconoce que ambas plataformas son suficientes para el objetivo propio del proyecto de fin de carrera y que la implementación de srsRAN parece más didáctica y accesible para su comprensión. Con esos considerandos se elige implementar esta maqueta de 5G con OpenAirInterface.

Capítulo 4

Implementación

Los dos objetivos siguientes del proyecto son implementar una red 5G de pruebas y escribir sus manuales de instalación y uso. Las decisiones tomadas y el trabajo de implementación realizado se describen en este capítulo. Los manuales se encuentran en el Apéndice A.

4.1. Core Network

El sistema OAI incluye como núcleo de red un Enhanced Packet Core (EPC) para 4G y 5G NSA, y un Core Network 5G (CN5G) para 5G SA. El grupo de proyecto OAI-5G Core Network (CN) trabaja en el desarrollo de un Core Network completo siguiendo la arquitectura definida por el 3GPP en TS 23.501 [5].

Se puede elegir instalar directamente sobre el hardware o en máquinas virtuales, se puede hacer un despliegue automático en contenedores Dockers, o se puede instalar en un ambiente nativo en la nube, utilizando Helm Chart (un cluster de OpenShift/Kubernetes). También se puede elegir la función de plano de usuario (UPF) a utilizar. La UPF hace la retransmisión de todo el tráfico entre el plano de usuario y las redes externas. Esto es conceptualmente simple, pero pueden ser cantidades enormes de tráfico, a los que no se les debe agregar limitaciones de ancho de banda ni aumentos de retardo relevante. La opción es entre SPGWU, una versión con capacidades de tráfico limitadas que carga poco el sistema, VPP-UPF, una versión de mejores prestaciones, construida utilizando un desarrollo de GTP-U de VPP hecha con Data Plane Development Kit (DPDK) [10], y una tercera opción de UPF de grado de producción utilizando SD-Fabric, un desarrollo open source de SDN, utilizando switches “white box” como plano de datos. Esta última opción aún está en desarrollo y requiere hardware específico. La opción VPP-UPF fue liberada en el transcurso de este proyecto, cuando el CN ya estaba en uso. Recomienda utilizar un servidor independiente y no fue probado en este trabajo. Parece un área interesante para trabajos futuros centrados en GTP-U y la interacción entre la red móvil y las redes de transporte. La opción del SPGWU utiliza un plano de usuario que es resultado de la modificación de la función equivalente en el EPC de 4G que existe en OAI. Para este trabajo, que se enfoca en la RAN, y no en

Capítulo 4. Implementación

el CORE, y cuyas pruebas se configuran con un único gNB y hasta tres UE, no impone ninguna restricción. En estas condiciones, desplegar Dockers con la opción de SPGWU como UPF es la opción implementada en este proyecto.

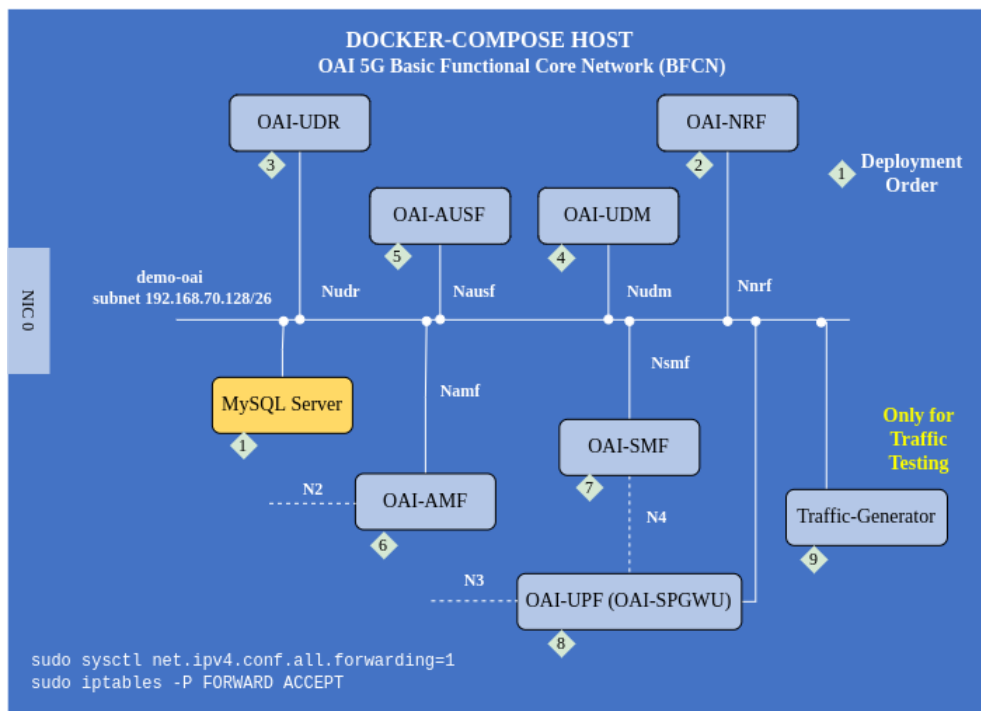


Figura 4.1: Diagrama de bloques del Core Network OAI instalado [14].

Este despliegue se puede correr solo con las funciones elementales, o con una versión básica, con la mayoría de las funciones, como se muestra en la figura 4.1. Para el uso previsto, se elige acceder a la configuración básica de las funciones de red del CN a través del archivo docker-compose.yml, a la base de datos de usuarios para incluir los UE a autenticar y a los logs de las funciones para analizar su comportamiento y resolver los problemas que se presenten.

4.2. Radio Access Network

La RAN puede instalarse en forma directa, desde el git oai-5g-ran-project o en un ambiente virtualizado desde el git mosaic5g. Esta última opción está enfocada en el desarrollo de RAN Intelligent Control, orquestadores y Software Deployment Kit (SDK) para el despliegue del sistema, y de las interfaces y el framework para utilizar xApps (aplicaciones externas) de ML/AI para el control y optimización del uso de los recursos en la RAN en los múltiples escenarios que habilita 5G. Para permitir posteriores investigaciones y desarrollos en esta línea de trabajo, es necesario partir de un conocimiento profundo y un manejo fluido de la infraestructura de base, la Radio Access Network implementada en el primer git. Adquirir esas capacidades es la razón de ser de este proyecto y el primer git es la opción tomada.

4.2. Radio Access Network

La RAN a su vez, tiene varias alternativas. El gNB puede utilizarse como un solo bloque, o dividido en Central Unit y Distributed Unit. Visto que la división está hecha para los ambientes virtualizados, concentrando las CU en EDGE Data Centers, y que su desarrollo es más reciente y parcial, se elige trabajar con gNB en un solo bloque.

El segundo foco de interés es utilizar efectivamente la interfaz de radio. Para ello la instalación de la maqueta se hace en dos computadoras, de modo de separar el gNB y el UE en equipos diferentes, conectados por la red 5G. En la misma computadora que corre el gNB, funciona el CN sobre la red de Dockers. Los enlaces de radio se establecen con SDR de alta capacidad que se conectan a las computadoras mediante una interfaz Ethernet de 10 Gb/s. El sistema radiante está implementado con antenas log periódicas de banda corrida, o con conexiones con cables coaxiales con atenuadores, cuando se quiere evitar recibir y generar interferencias. En la figura 4.2 se representa la red implementada.

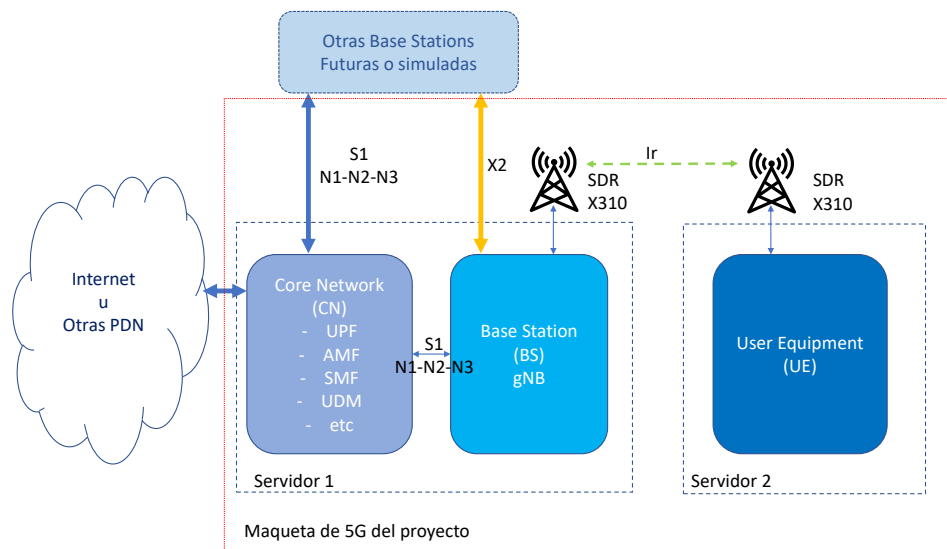


Figura 4.2: Diagrama general de la maqueta implementada, con Core Network y gNB funcionando en una computadora con un SDR como interfaz de radio y un terminal de usuario en otra computadora, también con SDR.

La maqueta básica incluye variantes con más computadoras. Utilizando SDR B210, conectados por USB 3.0 es posible agregar otros usuarios a la misma radio base. También se ensayó instalar el gNB en la segunda computadora, conectado al CN por red sin inconvenientes.

Además de la interfaz de radio es posible conectar el gNB con el UE por una interfaz de red que simula en enlace de radio. Esto se hace con la herramienta incluida en el sistema OAI, conocida como *RF Simulator* (RFsim). Este software reemplaza la placa de radio por comunicación TCP/IP para hacer posibles todas las pruebas funcionales sin equipo de radio. El OAI gNB y el OAI UE se comunican como si hubiera una interfaz de radiofrecuencia entre ellos. Las muestras I/Q que

Capítulo 4. Implementación

se envían y reciben normalmente a un SDR se pueden transmitir a través de esta aplicación directo entre el UE y el gNB. RFSim también es compatible con MIMO. Funciona como un servidor en el gNB, esperando los pedidos de conexión de los UE.

4.3. Especificación del hardware

A partir de los requisitos dados por las dos plataformas de software previamente estudiadas, se especificó el siguiente hardware. El mismo es suficiente para cualquiera de los dos sistemas en su estado de desarrollo al momento de la elección.

4.3.1. Servidores

Se utilizan 2 servidores con siguientes características, cuyas especificaciones técnicas son:

- Procesador 12th Gen Intel Core™i7-12700
12 Cores Intel®Integrated Graphics Frecuencia: 2.1- 4.9 GHZ Cache Size: 35 MB Threads: 20
- Placa Madre, con socket para el anterior procesador, con puertos vga, hdmi, usb y usbc
- RAM: 16 GB, DDR5
- Almacenamiento: 512 GB SSD
- Gabinete y fuente de poder acorde a las especificaciones anteriores
- Monitor, mínimo 17 pulgadas, vga o hdmi
- Recursos gráficos proporcionados por el procesador y la placa madre. No se requiere tarjeta gráfica especial.
- Sistema operativo Ubuntu 22.04.1 LTS, Kernel 6.2.0-33-generic.

4.3.2. Conexión con USRP

Con las características anteriormente descritas, se garantiza el nivel de procesamiento requerido. A este hardware hay que agregarle la conexión con el USRP X310 de 10 Gb Ethernet. Para ello se utilizan 2 tarjetas de red con las siguientes características:

- 1 tarjeta PCI express 10 Gbps Dual Port Sfp+ Pcie x8 Nic
- 1 tarjeta PCI express 10 Gbps Sfp+ Pcie x8 Nic
- 2 10 Gigabit Ethernet Cable w/ SFP+ Terminations

4.4. Opciones de Configuración

4.3.3. Interfaz de Radiofrecuencia

El gNB y el UE, se implementan utilizando los siguientes dispositivos SDR en la interfaz de radio de frecuencia, especificaciones en A.1:

- 2 SDR Ettus USRP X310 (ver en figura 4.3)
- con 2 UBX 160 daughterboard (una en cada uno)
- 3 SDR Ettus USRP B200
- 6 Antenas LP0965 (dos por SDR)
- 6 Cables coaxiales SMA-SMA
- 2 atenuadores de RF de 30 dB

4.3.4. Backup de Energía

- UPS 1000 KVA APC smart 1000
- 4 Cable de alimentación para UPS Características IEC 60320 C14, macho



Figura 4.3: Frente y fondo del SDR X310, indicando las interfaces de conexión utilizadas.

4.4. Opciones de Configuración

gNB

El sistema permite gran flexibilidad a partir de opciones de configuración. Los parámetros se asignan en un archivo de configuración al que se refiere en la invoca-

Capítulo 4. Implementación

ción del programa del gNB. Los mismos parámetros pueden establecerse directamente desde el comando de inicio. Tienen valores por defecto que en la mayoría de los casos permiten completar una configuración coherente. La consistencia entre los parámetros y la configuración buscada es responsabilidad del usuario. Errores en la consistencia es causa de que el software no inicie. Las configuraciones implementadas y eventuales mensajes de error son indicados en el log de la salida estándar.

El archivo de configuración utiliza formatos ASN-1 para representar las clases de definiciones que incluye. A modo de ejemplo y sin ser exhaustivo en la categoría gNB se definen:

- lista de PLMN admitidos, con MCC, MNC, y NSSAI; es necesario que coincidan con las definiciones del CN,
- tracking area code, que debe coincidir con el definido en el CN,
- dirección IP del AMF,
- dirección IP por la que se comunica el gNB con el CN,
- tiempo mínimo de respuesta de tx a rx (ej., PDSCH a PUCCH) medido en slots, es 5 para OAI UE y 2 para UE comerciales,
- ubicación en frecuencia del bloque de sincronización SSB y de la primera subportadora (PointA), expresados en ARFCN,
- la banda de frecuencia, el ancho de banda, la separación entre portadoras en el caso FDD,
- configuración TDD: número de slots de UL, de DL y definición de uso de los símbolos en slots híbridos, más el largo de la secuencia a repetir.

El archivo contiene también toda la configuración de radio, de los SDR a utilizar, define las Bandwith Parts, parámetros de los distintos protocolos utilizados. Se incluye un archivo de configuración completo y comentado en el manual entregado en el Apéndice A.3.2.

COTS UE

Para la conexión de un teléfono comercial es necesario utilizar una SIM programable, con el PLMN, el IMSI del usuario y las claves de acceso a la red. Para el proyecto se usaron SIM y una grabadora de SIM adquiridos a *Open Cells Project* [42], de Francia. También se usó el software de código abierto compartido por ellos para grabarlas (figura 4.4).

Se conectaron a la plataforma teléfonos móviles comerciales. Se obtuvo una conexión exitosa con teléfonos Samsung A33. Estos móviles poseen sistemas operativos Android en las versiones 12 y 13. Para ello se generó un nuevo archivo de configuración del gNB x310 con 217 PRB, aumentado las atenuaciones en recepción y así evitar que los móviles saturen el gNB. Además el parámetro *min_rxttime*

4.4. Opciones de Configuración

del gNB que habitualmente se configura con el valor 6, para lograr conectividad en los móviles se le asigna el valor de 2, o sea una importante reducción de este tiempo mínimo entre recepción y transmisión.



Figura 4.4: Las tarjetas SIM utilizadas en el proyecto, junto con la grabadora SIM USB.

El trabajo de configuración de móviles tiene una dosis importante de ensayo y error a través de ajustes de los diferentes parámetros. Una vez lograda la configuración estable, la misma queda validada para ese tipo de móvil específicamente, en este caso el Samsung A33.

UE OAI

Para ejecutar el UE de OAI en la segunda PC se deben pasar como parámetros en el comando de inicio los datos de IMSI, MNC, MMC y claves de acceso. También se pueden pasar en un archivo indicado en el comando. Un ejemplo de archivo de configuración de UE, `ue.conf` se adjunta en el manual de usuario en el Apéndice A. El UE OAI no hace una búsqueda por todas las bandas, como un teléfono comercial, y se le debe indicar banda, frecuencia central, ancho de banda y ubicación del SSB, para que encuentre al gNB e inicie el proceso de acceso del *Physical Random Access Channel* (PRACH). También se le deben dar los datos de conexión IP al SDR utilizado, por defecto busca un SDR en los puertos USB. Las capacidades del UE están definidas por defecto. Se pueden cambiar editando un archivo en formato

Capítulo 4. Implementación

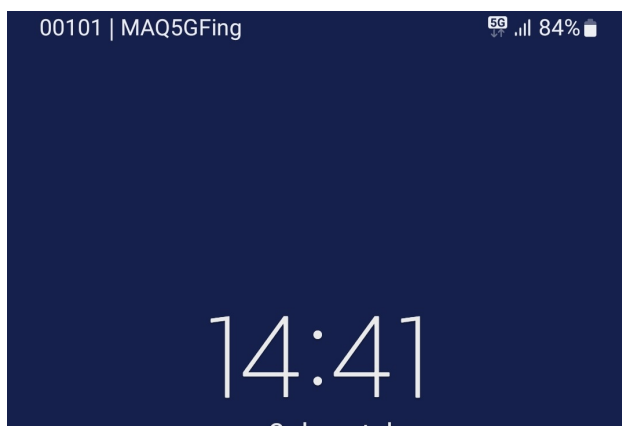


Figura 4.5: Celular samsung A33 conectado y registrado, en la parte superior se muestra el MCC, MNC junto al nombre de la maqueta y el logo del 5G.

XML e indicando su inclusión al ejecutar el UE. Esta operación es necesaria, por ejemplo para habilitar más capas y trabajar con MIMO con dos o cuatro antenas. Se incluye un archivo de capacidades de ejemplo con dos puertos de antenas en el manual citado.

Definiciones

La variedad de opciones exige tomar algunas decisiones generales para acotar los casos a utilizar. Las opciones se tomaron en base a tres criterios:

- comenzar con configuraciones simples,
- usar casos en RFSim replicables con las interfaces de radio disponibles,
- y avanzar con cambios parciales individuales que permitan comparar las consecuencias de los cambios.

Con configuraciones simples nos referimos a usar configuraciones básicas completas, dejando para proyectos futuros el uso de Carrier Aggregation (CA), múltiples Bandwidth Parts (BWP), y la implementación de varias *slices*, por ejemplo. También se limitan a configuraciones *Stand Alone*. No se probaron en este trabajo configuraciones *Non Stand Alone* (NSA) con red de núcleo 4G (EPC). Tampoco se probó el *IP Multimedia Subsystem* (IMS) para servicios de llamadas del tipo *Voice over New Radio* (VoNR).

El segundo criterio es acotar las decisiones en el entorno de simulación a casos replicables o comparables utilizando la interfaz de radio. En este caso, el hardware disponible acota la configuración de la interfaz de radio a una sola capa en el escenario principal, por disponer de una sola salida de radio en cada SDR. Eso hace utilizar en este proyecto, una configuración Single Input Single Output (SISO) en la generalidad de los casos. También se quiere utilizar antenas con transmisión por aire en un ambiente interior. Eso impone identificar bandas de frecuencia licenciadas, que no estén en uso para poder probar sin interferencia excesiva. El uso

4.4. Opciones de Configuración

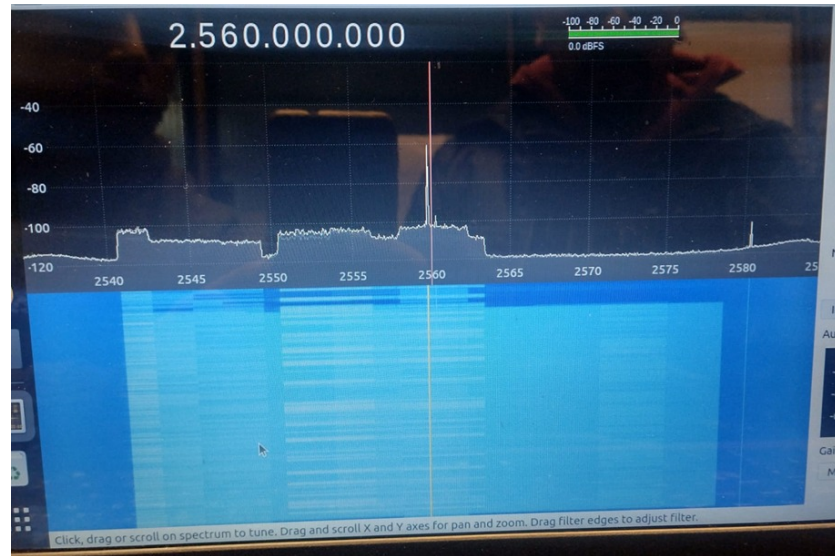


Figura 4.6: Espectro de un gNB con 40 Mhz de ancho de banda en la banda n41.

de TDD permite simplificar la búsqueda al no requerir de dos bandas apareadas. Esta decisión se refuerza por el interés que despierta esta técnica que predomina en 5G y no era utilizada en nuestra región por las generaciones anteriores. El uso de bandas no licenciadas (New Radio Unlicensed, NR-U), requiere de técnicas de “listen before talk” para convivir con Wi-Fi y otros usos, que aún no están implementadas en OAI.

En tercer lugar se busca que las configuraciones sean equivalentes en todos los parámetros, excepto en los que se prueban variaciones, y sus cambios relacionados. Como ejemplo, en las pruebas de performance, se comparan resultados variando el ancho de banda de un sistema sin otros cambios que no deriven de este.

A partir de la asignación de espectro para IMT hecho por la URSEC se definió hacer las pruebas de aire en la banda n41, en el rango entre 2540 MHz y 2630 MHz, que corresponde al salto intermedio de la banda 7 FDD de LTE. Se comprobó que no hay señales interferentes importantes con analizador de espectro implementado con los mismos SDR. En la figura 4.6 se ve un gNB en el rango seleccionado. También se identificaron rangos no utilizados todavía entre 3300 y 3600 MHz y por encima de 3800 MHz. Todas las pruebas de aire se realizaron con baja potencia, en ambiente interior y con tiempo de uso limitado. Fuera de estas frecuencias, las pruebas entre SDR fueron hechas con conexiones coaxiales entre los dispositivos.

Las pruebas con teléfonos se realizaron en tramos sin servicio de las bandas n77 y n78, debido a que en la banda n41 TDD no se logró la detección del gNB por parte de los teléfonos disponibles.

4.5. Escenarios de Uso

Se implementan varios escenarios, algunos con enlace simulado y otros con enlace de radio. En los simulados, el canal de radio se reemplaza por sesiones de RF Simulator, una aplicación que implementa una conexión IP entre el gNB que oficia de servidor y los UE que se conectan a él. Esta aplicación permite agregar un modelo de canal, con el que se agrega ruido AWGN. Los atenuadores programables en el gNB y el UE permiten completar las condiciones del enlace.

Los escenarios con dos SDR o con un SDR y un teléfono, requieren resolver en forma satisfactoria el balance del canal de radio para lograr establecer conexión. A su vez pueden dividirse en canales por aire y canales con RF conducida por coaxiales. Los enlaces por antena están más afectados por interferencia en recepción y a su vez, generan interferencia. Esta condición varía con la banda utilizada, y conviene verificar el medio con analizador de espectro. Al trabajar con enlaces por coaxiales se deben conectar en los transmisores atenuadores fijos para evitar daños en los SDR. Esto protege a los transmisores de señales reflejadas y a los receptores remotos, asegurando que la potencia máxima recibida es tolerable. Se utilizaron atenuadores de 30 dB en cada transmisor. Las pérdidas por transmitir por aire, la ganancia de antenas o el uso de atenuadores fijos, más el uso de atenuadores programados en los SDR y la definición de la máxima ganancia del receptor permiten establecer un correcto balance del canal de radio.

4.5.1. Sistemas con RFsimulator

Todo en Uno

En este primer escenario, todos los nodos se ejecutan en la misma máquina. Solo se puede usar con la interfaz de *RFsimulator* (RFsim). Es un escenario de gran utilidad para trabajos individuales sin grandes requerimientos de performance. Es óptimo para hacer capturas de tráfico en la interfaz del Docker a CN, y para validar configuraciones en distintas bandas, anchos de bandas u otros cambios, antes de probarlos con interfaces de radio.

Con CN, gNB y UE ya instalados, se inician las tres aplicaciones en una sola máquina. Con la versión de CN 1.5.1, se tiene una base de datos instalada, con los IMSI definidos del 001010000000001 al 001010000000004. Se comienza corriendo el CN y se debe esperar a que todos los dockers estén saludables (*healthy*). Es de interés correr los logs de las funciones del CN que interesen seguir. Las más útiles desde el punto de vista de la RAN son AMF y SMF. En la consola de AMF que se ve en la figura 4.7, se muestra la interacción con el UE y el gNB, se actualiza periódicamente el estado de conexión de los gNB existentes y de los UE conectados en cada uno de ellos. Durante los procesos de conexión y desconexión de un gNB o un UE, se registran en el log todos los mensajes de los protocolos involucrados con ellos y con las demás funciones del CN, lo que los hace una herramienta de análisis poderosa, aunque muy densa. La consola del SMF muestra la interacción del DHCP y de los demás procesos vinculados a la gestión de sesiones.

Primero se debe correr el gNB con la opción `--rfsim` incluida para habilitar

Capítulo 4. Implementación

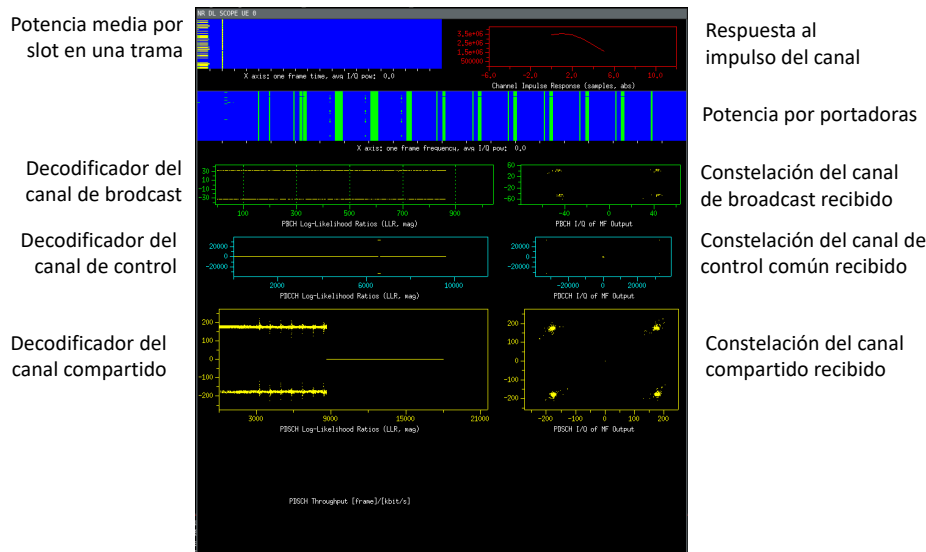


Figura 4.8: Scope de un UE mostrando las señales recibidas en Down Link.

Este escenario de pruebas tiene una variante importante que consiste en correr varios UE. El sistema por defecto viene configurado para un máximo de 16 UE por celda. Cada instancia de UE crea una interfaz con el nombre `oaitun_ue1`, que permite a la maquina host navegar utilizando la red de 5G. Si se ejecutan dos instancias en la misma PC, hay un conflicto por el nombre que no le permite iniciar a la segunda instancia. Para hacer pruebas con varios UE, sin necesidad de agregar más PC, pueden crearse *network name spaces* diferentes, y correr cada instancia en distintos *namespace*. El análisis más detallado del *scheduler* realizado, que se explica en el Capítulo 6, se pudo obtener en estas condiciones. En el manual de configuración del Apéndice A.2.6 se detalla como realizarlo. Otra opción es correr los UE y el GNB en dockers. Esto es posible y también se ejemplifica en el manual de configuración en el Apéndice A.3.1. Es una forma de aplicación cómoda para algunos usos y limitada para otros. La ejecución de la capa física del protocolo, es la parte más demandante del software. El sistema maneja por su cuenta los hilos de trabajo de los distintos cores, dejando un core reservado para la capa física. Al intermediar con el hardware mediante Dockers, y eventualmente correr varias instancias, en la misma máquina, se pierde eficiencia en el proceso. Por otro lado cuando el interés es modificar el código, se debe reconstruir la imagen del docker en cada compilación, lo que no es práctico. Finalmente, editar configuraciones mediante la consola de acceso externo al docker, es también más engorroso.

4.5.2. Sistemas con SDR

En estos escenarios se utiliza un SDR como radio del gNB y otro en el UE. En la figura 4.9 se muestra la maqueta funcionando. Se usaron los modelos X310 y B200, ambos en las dos funciones. El primero permitió hacer conexiones con 10

4.5. Escenarios de Uso

MHz, 40 MHz y 80 MHz de ancho de banda. Con los segundos solo se obtuvo el caso de 10 MHz.

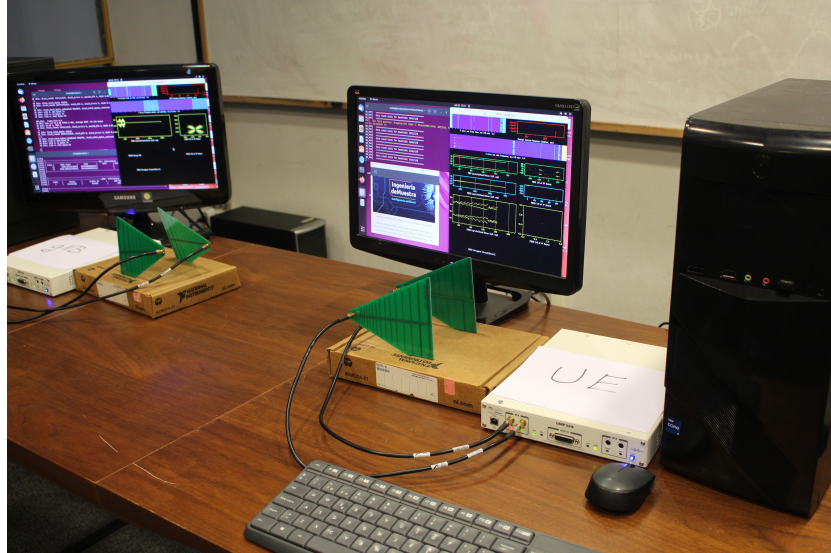


Figura 4.9: Maqueta en operación con dos SDR X310.

Dado que se dispone de dos X310 y varios B200, se usaron los primeros para las pruebas de una única conexión. Y se usaron varios B200 como UE para pruebas con múltiples UE equivalentes. También se utilizó un B200 adicional como analizador de espectro.

Otros modelos de SDR disponibles en Facultad, fueron probados, pero se descartaron por no cumplir con algunos requisitos. Los HackRF ONE y los ADALM-Pluto no tienen su driver incorporado al sistema de OAI. Los B100 de Ettus con 16 MHz de ancho de banda y muestras de 8 bits no llegan a las características mínimas requeridas, además de estar ya discontinuados y fuera de soporte lo que limita su exploración al esfuerzo propio. También se dispone de dos Ettus USRP N210 que no puede usarse porque su reloj interno permite un muestreo a 100 MS/s, pero con reloj de frecuencia fija de 100 MHz, no ajustable a múltiplos de la frecuencia de muestreo requerida. Los escenarios utilizados agregaron el radio a los hechos en modo conexión simulada y no requieren mayor descripción.

Escenario con Teléfonos Comerciales

Una variante al uso de un escenario con dos SDR es utilizar un gNB con SDR y teléfonos comerciales como UE. Configurando el gNB en una banda que el teléfono reconozca, utilizando un SIM con un IMSI registrado en el CORE y con los key adecuados para su autenticación es posible utilizar los servicios de la red creada por parte del teléfono. Se debe disponer de un acceso a internet en la interfaz de red del CN para la navegación.

No todas las marcas de teléfono ni todos los modelos se logran conectar a esta plataforma. El uso de hardware no dedicado y la ausencia en nuestro caso,

Capítulo 4. Implementación

de una fuente de sincronismo estable de referencia, generan jitter y corrimientos de frecuencia que no son toleradas por algunos dispositivos. Se conectaron a la maqueta dos teléfonos Samsung A33 5G. No fue posible conectar un Xiaomi Redmi 11 5G y un Samsung A22 5G.

Para futuros ensayos en este tipo de escenario, puede ser interesante incorporar a la maqueta un reloj GPS y una Caja Faraday. El reloj habilita instalar nodos MIMO usando más de un SDR y hacer pruebas con más de un gNB simultáneo. La Caja Faraday permite establecer un enlace controlado, libre de interferencias en cualquier banda entre el teléfono de pruebas y el sistema. Finalmente es interesante contar con un teléfono con clave de root para su administración con acceso al stack de protocolos y forzado de frecuencias en uso. Los módulos 5G Quectel son una alternativa muy utilizada en trabajos publicados.

4.6. Presupuesto

Se presenta el costo efectivo de la maqueta implementada. Se utilizan los precios reales de los componentes de hardware en condición plaza Montevideo. Las herramientas y sistemas de software utilizados son libres y sin costo. Las horas de dedicación se incluyen sin valorizar, dada la característica académico curricular de la dedicación. No se incluyen las horas docentes.

Presupuesto de Hardware (USD), Plaza, IVA incluido			
Ítem	Cantidad	Precio unitario	Subtotal
Computadoras Plaza	2	799,00	1.598,00
Monitores y accesorios	2	140,00	280,00
Tarjeta de red 2 x 10 Gb/s)	1	1039,00	1039,00
Tarjeta de red 1 x 10 Gb/s	1	380,00	380,00
Cables 10 Gb/s	2	128,00	256,00
UPS	1	1.013,00	1.013,00
SDR Ethus USRP X310	2	8.868,00	17.736,00
SDR Ethus USRP B200	3	1.924,00	5.772,00
UBX 160 daughterboard	3	1.620,00	4.860,00
Antena LP0965	6	90,00	540,00
Cables coaxiales SMA-SMA	6	75,00	450,00
Atenuadores SMA de 30dB	2	9,00	18,00
4 SIM y grabadora	1	43,00	43,00
Teléfono Samsung A33 5G	2	399,00	798,00
		Total	34.783,00

Tabla 4.1: Costo del hardware utilizado en este proyecto.

Con los recursos de hardware listados en la tabla 4.1 es suficiente para replicar este trabajo. Las horas de dedicación de la tabla 4.2 representan el esfuerzo requere-

Horas de dedicación.			
Rol	Cantidad	meses	Horas
Estudiantes	3	15	1.730

Tabla 4.2: Horas de trabajo empleadas en la realización del proyecto.

Software utilizado		
Ítem	Cantidad de Instancias	Precio unitario
Ubuntu 22.04	3	Sin cargo
Driver USRP 4.4.0	3	Sin cargo
OAI Core Network 1.5.1	2	Sin cargo
OAI RAN develop w42-2023	3	Sin cargo
Wireshark	3	Sin cargo
iperf3	4	Sin cargo
UICC/SIM programing v3.2	1	Sin cargo

Tabla 4.3: Listado del software utilizado en la implementación definitiva de la maqueta.

ruido y el tiempo que insume la curva de aprendizaje necesaria para esta tarea. Es válida para estudiantes de grado sin experiencia previa con el sistema utilizado y conocimiento acotado en la tecnología. Finalmente el listado de software indicado en la tabla 4.3 incluye los paquetes globales utilizados, aunque reconocemos en ellos todos los sistemas de código abierto incluidos como partes componentes de los mismos. Otras herramientas de software adicionales se usaron intensivamente para documentar o manejar la información, pero no son parte del sistema y no se listan.

Hemos descrito la maqueta implementada, las decisiones de diseño, y las formas de uso que utilizamos. Se repasaron las operaciones básicas y esbozaron algunas de sus posibilidades futuras. A continuación evaluamos sus capacidades y detectamos las limitaciones del sistema elegido y las restricciones de esta implementación.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Evaluación de Performance

Este capítulo está dedicado a evaluar la maqueta, explorar sus capacidades y limitaciones buscando caracterizar el rendimiento de la solución. Es un aporte que pretende ser relevante para las consideraciones iniciales de quienes necesiten utilizarla en el futuro.

La primera evaluación consiste en verificar su funcionalidad básica. Se crea un usuario, se conecta a la red, se verifica su conectividad IP y se conecta con redes externas. En segundo lugar se busca evaluar la capacidad de tráfico de la red. Se mide el *throughput* máximo posible en una conexión y se compara con el máximo teórico esperado. Se evalúa el comportamiento cambiando las condiciones y parámetros de la configuración.

Finalmente se aportan observaciones cualitativas respecto a la estabilidad y las respuestas del sistema a las condiciones de estrés durante las pruebas, al rendimiento del hardware utilizado y a la utilización de herramientas auxiliares del propio sistema.

5.1. Pruebas de Conectividad

La primera prueba elemental es comprobar la conectividad del UE. Una vez que los logs del UE, del gNB y sus scopes respectivos muestran el link de radio establecido, se puede comprobar en el log del AMF todo el intercambio que lleva a la conexión del UE. A continuación se listan los gNB en línea y los UE detectados con su estado de conexión, estos se observan en la figura 5.1. Si está registrado, se puede comenzar a probar.

Análisis de Captura NGAP

NG es el protocolo en capa de aplicación en las interfaces N1, N2 y N3 entre el bloque AMF del Core Network y un gNB de la RAN y los UE asociados. Se las conoce como N group, y de allí NG. Corre sobre SCTP/IP. Transporta la señalización entre el CN y el gNB y transporta la mensajería NAS entre el AMF y cada UE. La interfaz y el protocolo están descritos en los TS 38.412 y 38.413 del 3GPP. En la prueba realizada (figura 5.2) se capturan los paquetes en la interfaz

Capítulo 5. Evaluación de Performance



Figura 5.1: Logs del core AMF, donde se muestran el gNB conectado y el UE registrado.

entre el gNB y el Core Network. En este caso corresponde a la interfaz demo-oai, definida entre los dockers del CN y la PC host, donde también corre el gNB. En la imagen se muestra la captura filtrada por NGAP. Se inicia la captura con solo el CN corriendo, se inicia un gNB y luego se inicia un UE.

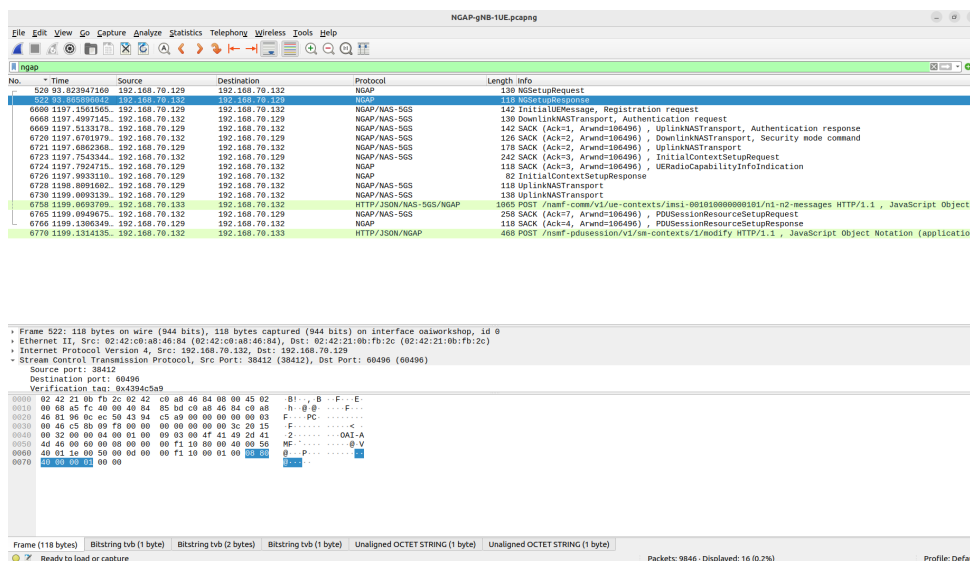


Figura 5.2: Captura de NGAP en conexión de un UE.

Los primeros dos mensajes corresponden al inicio de la sesión del gNB. Son un NGSetupRequest enviado por el gNB, con sus datos y su TAC configurada y un NGSetupResponse con todos los datos de la red soportada por el AMF (GUAMI, PLMN Item, nombre de AMF, funciones soportadas, etc.). Luego de registrado el gNB no hay más mensajes hasta que inicia un UE.

5.2. Pruebas de Throughput

El tercer mensaje corresponde al pedido para registrarse del UE, un InitialUE-Message. El AMF responde inicializando una capa de transporte NAS con este UE con un DownlinkNASTransport donde se piden datos para Autenticar el usuario. La secuencia de mensajes que sigue incluye el inicio de un UplinkNASTransport, y el set up del contexto del UE, con la consulta y respuesta de las capacidades del UE. Los datos de conexión IP del UE se transmiten encriptados y no se aprecian, pero el resultado de este proceso es un UE conectado, con IP asignada del rango manejado por el SMF.

Conectividad IP

Si la configuración es correcta, el UE debe de tener una IP asignada por el SMF del Core. Se puede verificar ejecutando el comando “ip a list” en una consola del UE y se comprueba que se ha creado una interfaz de nombre oaitun_ue1 con una ip de la red 12.1.1.0/24, rango definido en la SMF del CN. Si se agrega una ruta por defecto a la puerta de enlace 12.1.1.1 se puede establecer conexión con el CN y las redes externas a él conectadas. Se verifica la conectividad haciendo un ping a la dirección 192.168.70.135 que corresponde a una función auxiliar generadora de tráfico. Hay un servidor iperf3 corriendo en esa NF, por lo que es posible también generar tráfico mayor a esa interfaz.

Conectando la PC host del CN a internet, es posible hacer ping desde el UE a IP públicas conocidas y navegar por páginas web. Tener en cuenta las configuraciones DNS locales del dispositivo para no generar conflicto con los DNS del core, lo más recomendable es obtener estas configuraciones en forma automática. Además el core debe salir a internet de forma directa, evitando saltos a través de servidores proxy u otra configuración que aplique restricciones. También es posible hacer conexiones IP entre dos UE conectados.

5.2. Pruebas de Throughput

Las pruebas de performance están centradas en determinar el througput máximo de una conexión a través de un core conectado a un gNB y uno o dos UE simultáneos. Con este objetivo se definen dos conjuntos de pruebas utilizando el esquema cliente servidor de iperf3. En uno se harán pruebas utilizando la conexión RF Simulator (RFsim) figura 5.3, de un gNB con UE conectado, que se repiten variando el ancho de banda del canal. En este mismo set se realizan las mismas pruebas, utilizando un enlace de radio frecuencia con SDR, usando antenas (figura 5.4) y conexiones coaxiales con atenuadores(figura 5.5). En el segundo set de pruebas, se repiten las instancias anteriores en una configuración de 1 gNB al cual se conectan 2 UE (figura 5.6).

Se tiene la limitante de que los SDR X310 soportan un máximo de 217 PRB, 80 MHz. Con RFsim se llegó a probar hasta 273 PRB, que corresponde a 100 MHz, el máximo posible para una portadora en FR1 con SCS de 30 kHz. Todas las configuraciones usadas son de una sola portadora (1 CC), un solo puerto de antena (SISO), y con una bandwidth part igual al ancho de banda total de la celda.

Capítulo 5. Evaluación de Performance

El sistema usado es TDD, con un ciclo de 10 slots. Se asignan 7 slots de DL, 2 de UL y uno híbrido con 6 símbolos de DL, 4 símbolos de separación y 4 símbolos de UL.

El throughput es evaluado con servidores iperf3 externos e internos a la red OAI. El servidor externo se implementa en un host conectado por fuera de la red del core y un cliente corriendo en uno de los UE. El segundo caso (servidor interno), es implementado en uno de los UE y el cliente corriendo en un segundo UE (figura 5.7). Sirve para verificar si el Core Network y el UPF haciendo NAT a las redes externas, representan una limitación a las pruebas.

Primer set de pruebas gNB + 1 UE

Aire, gNB X310 + 1 UE x310:

- 24 PRB - 10 MHz
- 106 PRB - 40 MHz
- 217 PRB - 80 MHz

Conexión de los SDR por coaxial:

- 217 PRB - 80 MHz
- 106 PRB - 40 MHz

RFsim:

- 24 PRB - 10 MHz
- 106 PRB - 40 MHz
- 162 PRB - 60 MHz
- 217 PRB - 80 MHz
- 273 PRB - 100 MHz

Segundo set de pruebas gNB + 2 UE

Aire, 24PRB - 10 MHz - gNB x310 + 2UE B200:

RFsim 106PRB - 40 MHz:

5.2. Pruebas de Throughput

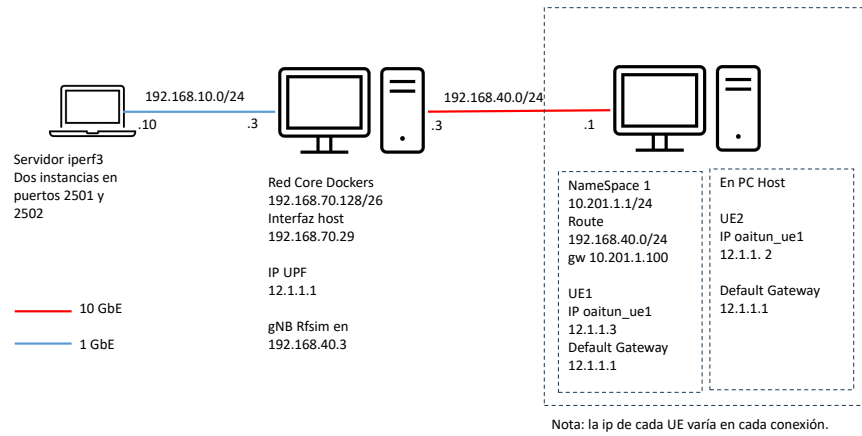


Figura 5.3: Esquema de de dispositivos y conexiones para pruebas de rendimiento con un gNB y UE conectados por cable 10 Gb Ethernet, utilizando RFSim.

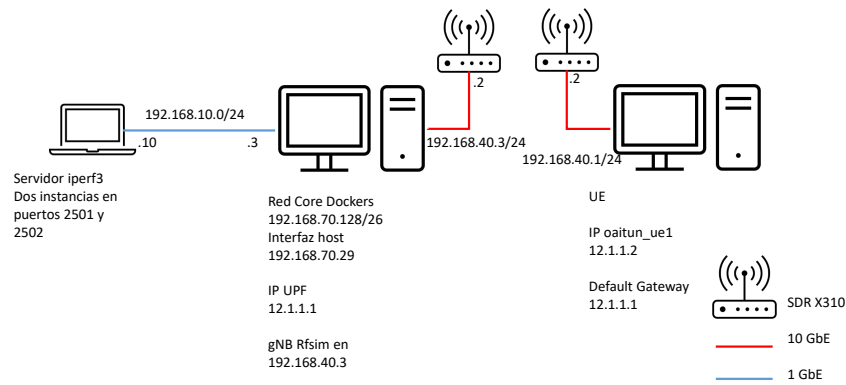


Figura 5.4: Esquema de de dispositivos y conexiones para pruebas de rendimiento con un gNB y UE en SDR con antenas.

Performance usando Radio Frecuencia por Aire y coaxial

En la figura 5.8, se muestra el throughput discriminado por medio: aire según esquema de conexión de la figura 5.4 o cable coaxial según esquema de la figura 5.5, ancho de banda según cantidad de PRB (24, 106 o 217) y en el sentido de la comunicación (UL o DL). Además se contrastan con los máximos teóricos de cada

Capítulo 5. Evaluación de Performance

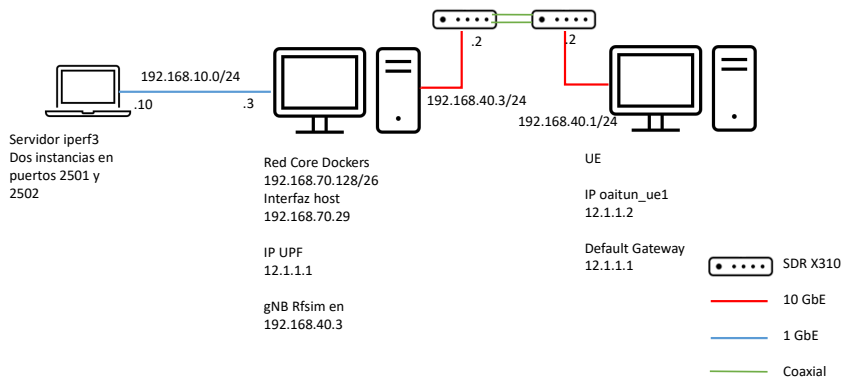


Figura 5.5: Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y un UE en SDR conectados por coaxial.

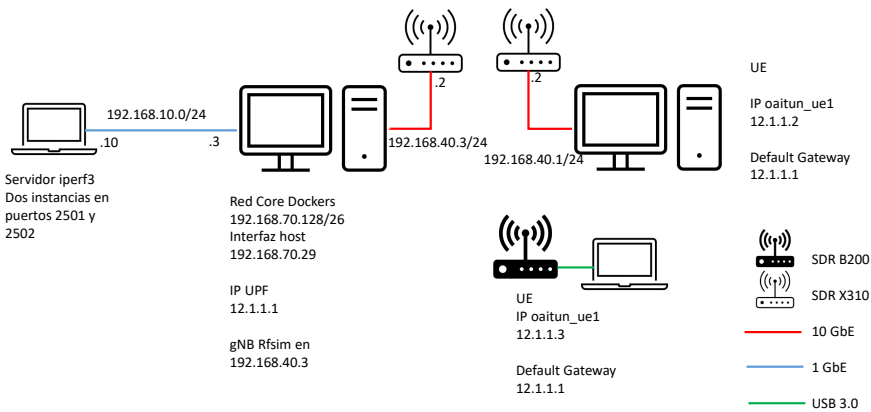


Figura 5.6: Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y 2 UE en SDR, y servidor Iperf en un PC externo a la red

categoría, calculados según la ecuación 5.1, en base a la especificación TS 38.306 [2]. El gráfico muestra que el rendimiento con 24 PRB es muy bajo, esto es consecuente con el ancho de banda que es el más bajo de esta configuración. A medida que aumenta los PRB es esperable un aumento en *throughput*, cosa que se observa en la gráfica. Por ejemplo en el aire este aumento es aproximadamente proporcional (DL/UL Mb/s): 7.73/4.7 a 24 PRB, 42.8/8.21 a 106 PRB y 63.7/12.5 a 217 PRB.

5.2. Pruebas de Throughput

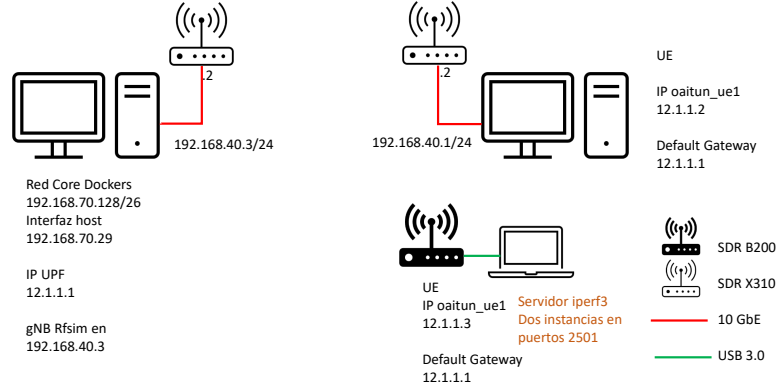


Figura 5.7: Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y 2 UE en SDR, servidor Iperf en uno de los UE o sea interno a la red.

En la conexión por coaxial se obtiene el throughput máximo de 75.1/8.26 (DL/UL Mb/s), que está muy lejos del que establece la teoría 258.7/79.8 (DL/UL Mb/s) para un enlace con condiciones óptimas.

Cálculo Throughput Teórico (Th):

$$Th(Mb/s) = 10^{-6} * \sum_{j=1}^j (v_{layers}^{(j)} * Q_m^{(j)} * f^{(j)} * R_{max} * \frac{N_{PRB}^{BW(j),\mu} * 12}{T_s^\mu} * (1 - OH^{(j)})) \quad (5.1)$$

- donde j representa la cantidad de componentes de las portadoras,
- $v_{layers}^{(j)}$ representa en numero de *layers* MIMO,
- $Q_m^{(j)}$ es el orden de modulación siendo: 2-QPSK, 4-16QAM, 6-64QAM, 8-256QAM,
- $f^{(j)}$ para ajustar a la realidad el pico de rendimiento en un determinado periodo de tiempo, teniendo en cuenta pérdidas por retransmisiones asignaciones etc. 0.4, 0.75, 0.8 y 1,
- R_{max} corresponde al número 948/1024 de la mejor tasa de codificación posible,
- $N_{PRB}^{BW(j),\mu}$ corresponde al número máximo de PRB,
- μ corresponde a la numerología del espacio entre subportadoras siendo: 0-15KHz, 1-30KHz, 2-60KHz,

Capítulo 5. Evaluación de Performance

- $T_s^\mu = \frac{10^{-3}}{14 \times 2^\mu}$ corresponde al tiempo promedio de duración del símbolo OFDM en una trama de la numerología μ ,
- $OH^{(j)}$ *Overhead*; tasa de bits utilizados por encabezados de protocolos, señales de medida y canales de control sobre el total disponible: 0.14 para DL en FR1, 0.18 DL en FR2, 0.08 para UL en FR1, 0.10 para UL en FR2.

Ejemplo de cálculo de ancho de banda para el caso de FR1, SCS de 30 kHz, 1 CC, SISO, orden de modulación 64 QAM y 217 PRB. Para este caso los parámetros de la ecuación 5.1 son los siguientes:

- $j = 1$
- $v_{layers}^{(j)} = 1$
- $Q_m^{(j)} = 6$
- $f^{(j)} = 1$
- $R_{max} \simeq 0,926$
- $N_{PRB}^{BW^{(j)},\mu} = 217$
- $\mu = 1$
- $T_s^{\mu=1} \simeq 3,57 \times 10^{-5}$
- $OH^{(j)} = 0,14 DL / 0,08 UL$

El ajuste final se realiza multiplicando toda la ecuación por el factor de ajuste de acuerdo a la distribución de los ciclos de slots en la trama. En el sistema TDD utilizado se asignan 7 slots de DL y 2 UL y un híbrido de 6 símbolos de DL, 4 de guarda y 4 símbolos de UL. Con esta configuración el factor de ajuste es:

- $DL = \frac{7 + \frac{6}{14}}{10} \simeq 0,743$
- $UL = \frac{2 + \frac{4}{14}}{10} \simeq 0,214$

Los resultados para el *throughput* en ambos sentidos para este ejemplo son:

$$Th_{DL}(Mb/s) = 10^{-6} (16 \times 1 \times 0,926 \times \frac{217 \times 12}{3,57 \times 10^{-5}} \times (1 - 0,14)) \times 0,743 \quad (5.2)$$

$$Th_{DL}(Mb/s) = 258,74$$

$$Th_{UL}(Mb/s) = 10^{-6} (1 \times 6 \times 1 \times 0,926 \times \frac{217 \times 12}{3,57 \times 10^{-5}} \times (1 - 0,08)) \times 0,214 \quad (5.3)$$

5.2. Pruebas de Throughput

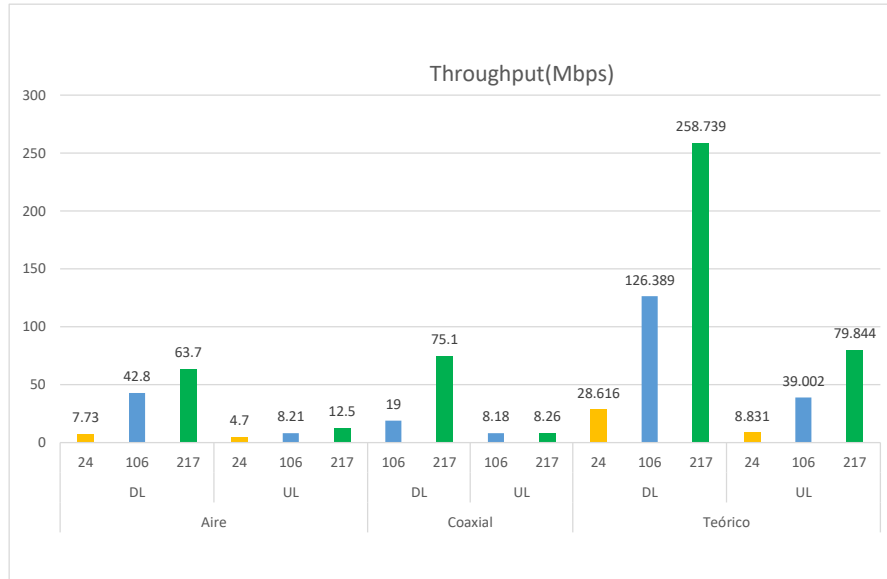


Figura 5.8: Throughput según medio y PRB, se comparan datos obtenidos de la transmisión por aire y coaxial, comparados con los máximos teóricos.

$$Th_{UL}(Mb/s) = 79,84$$

En los enlaces con antenas se observa un número alto de retransmisiones y un MCS que queda en valores intermedios como consecuencia de los errores. Por lo tanto, el índice de modulación Q_m y la tasa de codificación R son menores a los valores máximos tomados en el cálculo teórico, lo que junto a las retransmisiones explica la diferencia. El número de retransmisiones en primera y segunda instancia es reportado cíclicamente en el UE. El valor del MCS activo, y el valor de señal RSRP se muestran en el log del gNB para cada UE conectado, actualizándose cada pocos segundos. Este comportamiento es consecuencia de la alta variabilidad del canal en un entorno interior cerrado, con alto impacto del fading por caminos múltiples. A esto se le suma el uso de antenas diferentes en transmisión y recepción, mientras que el sistema evalúa el canal como si fuese simétrico al ser TDD. Por otro lado, en la conexión por coaxial se mantienen los parámetros estables pero sin llegar al MCS máximo y con algunas retransmisiones.

También se debe señalar que el balance del canal, es un proceso iterativo que parece más de ensayo y error, ya que no incluye medidas objetivas directas. Se ajustan por un lado los objetivos de señal ruido en UL y DL, la ganancia máxima de transmisión y el valor de dos atenuadores por software, que se implementan en el SDR para ajustar las condiciones del canal. El impacto del ajuste de los atenuadores no es visible en la medida del RSRP por el efecto del control automático de la potencia de transmisión. Si los atenuadores quedan muy bajos, se satura el

Capítulo 5. Evaluación de Performance

gNB y no se conecta el UE. Si son muy altos, no se detecta la red. El rango útil es acotado y varía sus valores con el ancho de banda y la frecuencia de trabajo. El resultado de este ajuste es determinante en el rendimiento de las pruebas, e impone valores de ajuste muy diferentes para un enlace con antenas que para uno por cables.

Por otro lado, un dato particular debe considerarse para los 106 PRB, que es la baja en el rendimiento cuando el tráfico es por coaxial, pasamos de 30.1 Mb/s en radio frecuencia a 19 Mb/s en coaxial. Estas particularidades, sumadas a que los rendimientos están muy por debajo de los establecidos teóricamente, se explican por el desbalance en los parámetros del enlace coaxial, recién explicados.

Performance con RF Simulator

Con RF Simulator (RFsim) disponemos de una herramienta de simulación que permite obtener datos en un ambiente controlado y con parámetros bien definidos como elemento de comparativa, es entonces que se evalúan todos los casos anteriores y se agregan los de 162 y 273 PRB que corresponden a anchos de banda de 60 y 100 MHz respectivamente, en base al esquema de la figura 5.3 . Esto último se muestra en el gráfico 5.9, lo primero a destacar es que los datos no se condicen con lo que establece el modelo teórico (figura: 5.8). El rendimiento está muy por encima del valor teórico en los anchos de banda menores (24, 106 y 162 PRB) y bastante por debajo en los mayores (217 y 273 PRB). Además surge una particularidad, el pico de rendimiento se da en DL en la configuración de 162 PRB con 279 Mb/s y no en 273 PRB como se podría esperar.

La causa de este comportamiento está en el uso del RFsim. Este es un simulador de enlace que intercambia muestras I/Q entre gNB y UE utilizando una conexión IP sobre ethernet, un enlace asíncrono que está funcionando con interfaces de 10Gb/s, que no impone ninguna restricción al intercambio. La velocidad de transferencia de datos no se hace pausada por la frecuencia de muestreo, sino que se hace a la velocidad impuesta por el procesamiento en los computadores. Si se procesan y transmiten más muestras por segundo que las pausadas, el throughput será mayor al esperado. Si se procesan más lento será menor de lo esperado, y todo sin que implique fallas en el funcionamiento.

El intercambio de datos en la interfaz de radio en 5G es plesiócrono, establecido por una tasa de muestreo fija en el transmisor que se mantiene incambiada independientemente de la cantidad de datos transmitidos, y una frecuencia de muestreo en el receptor recuperada de la señal recibida y en sincronía con el transmisor. El tiempo en el procesamiento del sistema OAI se calcula a partir del conteo en las muestras recibidas desde el SDR, según se explica en el GIT en la arquitectura del software [25]. A partir de este conteo se computan los tiempos correspondientes a un slot, (que es el tiempo de ejecución TTI del lazo infinito del software) y a una trama. La ejecución de las funciones de transmisión se disciplinan a este tiempo de ejecución.

Los osciladores de los SDR son de buena calidad, y tienen además la posibilidad de sincronizarse mediante un receptor de GPS y/o un distribuidor de sincronismo. Esta opción es recomendada por OAI y por su comunidad en los foros. Es una

5.2. Pruebas de Throughput

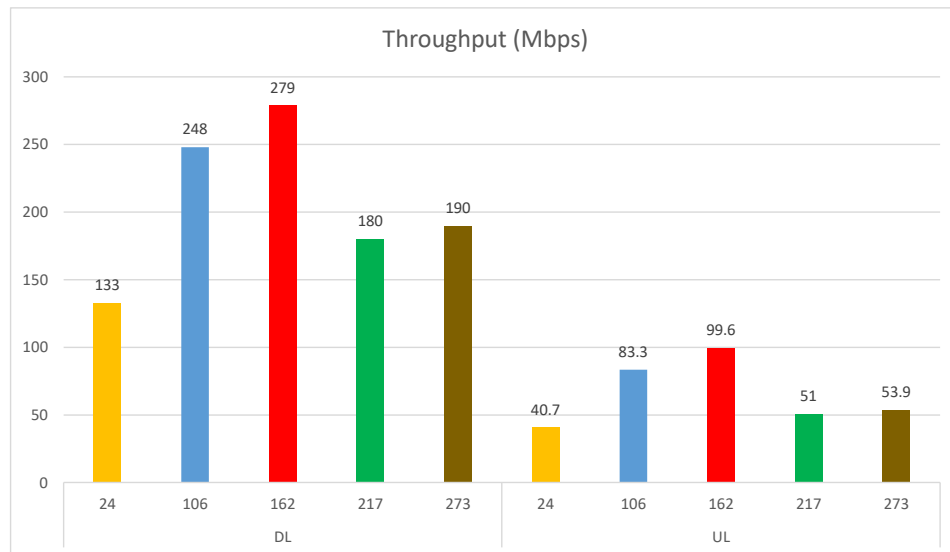


Figura 5.9: Throughput en RFSim Según PRB donde se muestran los datos obtenidos, en los que se observan rendimientos muy por encima de lo teórico para los anchos de banda más bajos.

mejora a la maqueta que vale la pena considerar en un futuro. Mensajes de Warning por ajustes en el corrimiento de trama son habituales en el la salida de consola del UE por radio. Y se ejecuta el programa del gNB con un comando de ajuste de sincronismo más laxo, para cuando no se dispone de señal de sincronismo externo. En probables escenarios futuros, con múltiples gNB, o escenarios NSA, con eNB y gNB, el sincronismo entre nodos pasa a ser imprescindible.

Cuando se utiliza RFSim, las muestras recibidas vienen de la otra PC por una conexión de red y no fueron pautadas por un oscilador de referencia. El tiempo en este escenario no tiene correspondencia con el tiempo real, y queda establecido por algún valor de equilibrio entre los procesos de ambas PC. Una observación que coincide con esta explicación surge de la figura que se tomó durante pruebas con RFSim, con 273 PRB. El porcentaje de uso de la red ethernet de 10GB/s debe ser constante ya que solo se envían muestras a una tasa fija, pero en los lapsos de tiempo en que los procesadores aumentan su trabajo la red baja (figura:5.10), lo que solo se explica con una menor velocidad de transmisión de las muestras.

Comparación de Tráfico entre 2 UE y Redes Externas

El segundo set de pruebas en lo referente a performance, tiene el objetivo de evaluar la plataforma con un gNB a cual se conectan dos UE. Se evalúan por un lado la conexión de ambos por aire en radio frecuencia, en este ítem debemos precisar que el hardware disponible para este caso son 2 UE implementados con

Capítulo 5. Evaluación de Performance



Figura 5.10: Comportamiento de la CPU y del tráfico ethernet durante la transmisión en RFSim.

SDR B200 y conectados a un gNB con SDR X310, siguiendo el esquema de la figura 5.6. Por esto último, esta prueba está configurada en el máximo soportado por B200, 24 PRB, 10 MHz en la banda n41. Se comparan el *throughput* de los 2 UE transmitiendo a la vez tanto DL como UL. También se compara los casos en el que el servidor iperf esta fuera de la red OAI (externo) y los casos donde el servidor iperf se encuentra en uno de los UE, o sea es interno a la red, configuración dada por el esquema de la figura 5.7. Estos datos se observan en la gráfica 5.11, donde se destaca la baja en el rendimientos cuando el servidor iperf se encuentra en el UE2, de 1.29 Mbps en DL, con respecto al mismo caso con servidor iperf externo a la red OAI de 5.63 Mbps. Estos resultados tienen lógica, la baja en el rendimiento con el iperf en el UE2 se explica por la interfaz aire-aire entre las dos conexiones limitando el UL el troughput en ambos casos. No hay evidencias de que la salida por el Core Network a redes externas imponga limitaciones para un solo UE. Para otros registros de prueba dirigirse al Apéndice C, Protocolos de prueba y resultados.

Otro aspecto que atañe a la evaluación de la performance, es cotejar el rendimiento de la maqueta en RFSim con 2 UE generando tráfico a la vez y por separado, a través del esquema de la figura 5.3. En el gráfico 5.12, se compara el rendimiento del sistema cuando transmiten los UE por separado y cuando transmiten los 2 a la vez. Todos los casos están referidos a 106 PRB en la banda n78. El pico de rendimiento se produce en DL para el UE2 en 149 Mbps transmitiendo por separado, pero el aspecto más destacable es que cuando ambos UE están trans-

5.2. Pruebas de Throughput

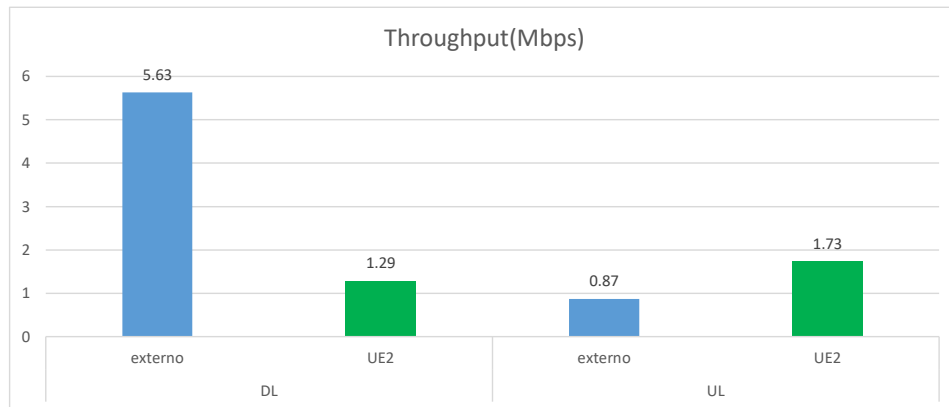


Figura 5.11: Throughput aire 2 UE, según ubicación del servidor iperf, interno (color verde) o externo a la red (color azul).

mitiendo, el rendimiento por UE se reduce casi hasta la mitad, además de que el ancho de banda se reparte en forma aproximadamente equitativa, 86.8 Mbps para el UE1 y 84.0 Mbps para el UE2. Algo muy similar ocurre en UL cuando ambos UE's están transmitiendo, 9.15 Mbps para UE1 y 8.78 Mbps para el UE2. Este comportamiento tendiente a mantener el mismo nivel de ancho de banda en ambos UE, se explica debido a que por defecto el algoritmo de scheduling es Proportional Faire (PF)

Se observa en todos los casos un mayor ancho de banda en el enlace descendente (DL) respecto al ascendente (UL). Esto es coherente con la distribución de los slot en la subframe, que en los archivos de configuración de gNB utilizados tiene asignado 7 slot para DL y 2 para UL. Por otro lado, si bien, un aumento de PRB tiende a un aumento en el *throughput*, consecuente al aumento del ancho de banda, este depende de las condiciones del medio y del correcto balance del enlace. Si es aire en radio frecuencia hay que considerar la interferencia, la propagación multicaminos, la distancia entre gNB y UE, etc. Si el medio es el esquema cable coaxial y atenuadores, se debe balancear correctamente, ajustando la configuración a través del archivo perteneciente a cada banda y PRB, en lo referente principalmente a las atenuaciones en transmisión/recepción (TX/RX) y potencia de TX/RX, tanto en el gNB como en el UE, para así evitar la saturación en caso de potencia excesiva o mala recepción en caso de potencias muy bajas.

En lo referente a RFSim, es una herramienta muy útil para caracterizar la maqueta en un ambiente simulado. Y es óptima para hacer análisis de protocolos en condiciones controladas. No es útil cuando la variable tiempo es determinante.

Capítulo 5. Evaluación de Performance

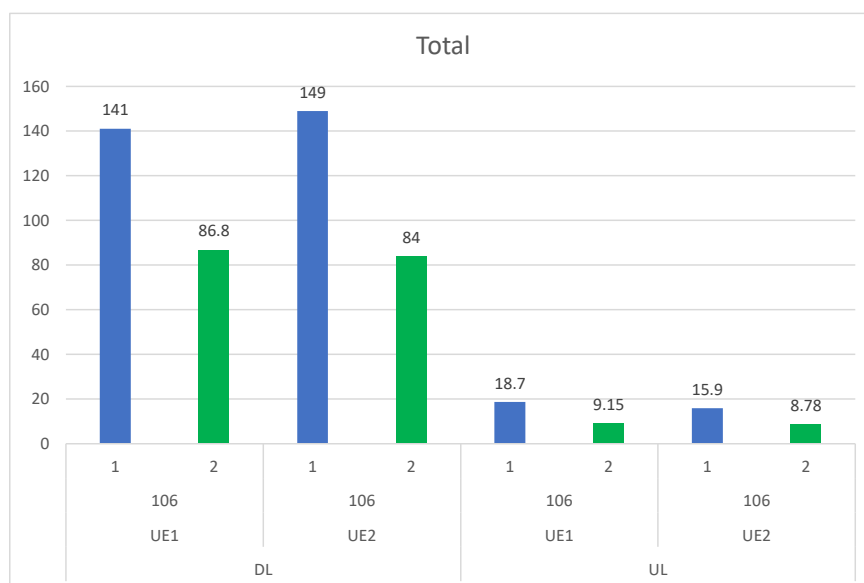


Figura 5.12: Throughput RFSim 2 UE, transmitiendo solos (color azul) y en forma simultánea (color verde).

Y lógicamente, tampoco sirve para estudiar condiciones de radio.

Conexión de Teléfonos Comerciales

Tal cual se expuso en el capítulo de implementación, se conectan a la red dos teléfonos móviles Samsung A33, con el gNB usando SDR X310 ejecutando la configuración de 217 PRB para móviles. Una vez registrados en el core, se hicieron test de conectividad hacia internet desde los móviles. Los resultados de la misma son congruentes con el tipo de conexión a internet del core (figura 5.13), verificando definitivamente que la traza de conectividad de los móviles es la correcta.

En cuanto a cobertura, con esta instalación en un ambiente interior conformado en su mayoría por paredes de vidrio y metal, se logra una conexión estable hasta los 40 metros. Como se especifica en el apéndice A, en la parte de datos de hardware A.1, las antenas utilizadas con los SDR son de periodo logarítmico y ganancia 6 DBi, una para recepción y otra transmisión.

En cuanto a estabilidad, el sistema fue evaluado durante una exposición con un flujo de personas constante y de alta densidad, con interferencia de señales 4G y 5G. La conexión de los móviles al gNB se mantuvo ininterrumpida durante las 3 horas de la muestra, cabe destacar que la misma fue en un ambiente cerrado de paredes de material tradicional y divisiones internas en yeso, con personas circulando en forma permanente. Es en ese contexto que se establecieron comunicaciones de vídeo llamadas con la aplicación google meet, la misma fue de muy buena calidad.

5.3. Consideraciones Cualitativas Sobre el Hardware y la Estabilidad del Sistema



Figura 5.13: Speed test de internet, de móvil samsung A33 conectado al Core a través de SDR X310. El core esta conectado al internet libre de la Facultad de Ingeniería

5.3. Consideraciones Cualitativas Sobre el Hardware y la Estabilidad del Sistema

En condiciones de funcionamiento normal, ambos PC funcionan con un uso holgado de memoria y suficiente capacidad de procesamiento. Al aumentar la demanda de tráfico, el uso de procesadores aumenta a valores muy altos, pero sin llegar al 100 % (figura 5.14).

Cuando la demanda de tráfico supera en mucho la capacidad del sistema, se producen funcionamiento anómalos. Si se hacen pruebas de iperf, en UDP, sin límite de velocidad, el troughput resultante es significativamente menor que si se ajusta la velocidad de transferencia del iperf al valor máximo esperado. En esas condiciones se maximiza el resultado.

Cuando esta prueba se repite con enlace de radio, suele suceder que aparecen múltiples mensajes de Service Data Unit (SDU) unavailable. Un mensaje que indica que se interrumpe un Transmission Time Interval (TTI) por no completarse el procesamiento en el tiempo máximo. Cuando esta condición es repetitiva, el enlace no se restablece y se pierde la conexión ip, aún después de cancelar la prueba en curso. Para salir de esta situación se precisa parar y reiniciar el UE.

Cuando no se lleva la capacidad de tráfico al límite no se han observado interrupciones inesperadas del software. Se ha mantenido un enlace funcionando por lapsos de varias horas sin inconvenientes.

Capítulo 5. Evaluación de Performance



Figura 5.14: Desempeño de la cpu en el servidor del core cuando la plataforma esta corriendo con un gNB y 2 UE conectados transmitiendo datos.

Capítulo 6

Desarrollo de un Scheduler Alternativo

Otro objetivo específico del proyecto es “Implementar un desarrollo de software que valide las potencialidades de la maqueta como entorno de desarrollo”. En este capítulo se selecciona la aplicación a implementar, se analiza la estructura general del sistema y de la aplicación a sustituir, y se describe la hecha. En el Apéndice D, se documenta el código implementado y se explica como compilar el código de OAI con cambios.

6.1. Selección de la Aplicación a Desarrollar

La opción de desarrollar un *scheduler* fue el planteo inicial propuesto en el tema. Visto que es un área temática con mucho trabajo acumulado en el Instituto de Ingeniería Eléctrica (IIE), donde se ha trabajado en simuladores y se investiga en vinculación con la aplicación de técnicas de ML/AI, es razonable hacer una primera aproximación a implementar código funcional en la maqueta vinculado a esta temática. Se busca sustituir el procedimiento actual del sistema por un algoritmo clásico alternativo, manteniendo la funcionalidad del sistema y evaluando su comportamiento.

6.1.1. Descripción de un *Scheduler*

El *scheduler* es el proceso de alto nivel en el en que se organiza la distribución del ancho de banda disponible entre los usuarios que lo demandan. Se distribuye en base a asignación de rangos de frecuencia y períodos de tiempo a cada usuario. Es un proceso que ocurre en la capa MAC, en interacción con la capa física, y por el cual se decide el orden de prioridades de la asignación. Para el DL en la capa MAC se hace el armado del transport block (TB) que se le entrega a la capa física (PHY) multiplexando todos los flujos de datos de usuario y control dirigidos a todos los UE en la celda (figura 6.1). Se hace adjudicando recursos para todos los procesos de control, mandatorios y opcionales, y luego se adjudican los recursos remanentes, multiplexando los datos de distintos flujos de datos según la prioridad establecida. Se le indica a la capa PHY cómo se debe configurar para transmitir ese

Capítulo 6. Desarrollo de un Scheduler Alternativo

TB y se debe enviar mensajes de control (Control Elements, CE) a los UE con la información de dónde, cuándo y cómo va la información que les corresponde. Esta información incluye modulación, Códificación, Resource Blocks y slots asignados. Un proceso similar se hace en sentido UL. Se setea el UE, se informa a la capa PHY dónde y cómo recibe la información.

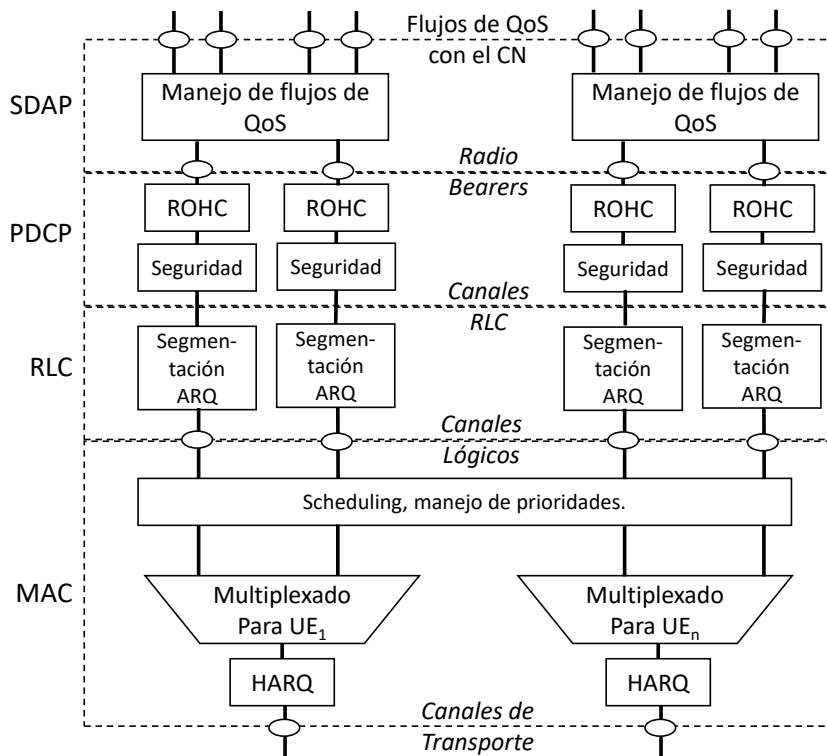


Figura 6.1: Multiplexado de datos de capa MAC en Down Link, para el armado del Transport Block [13].

Además de asignar frecuencia y tiempo, y contabilizar esos recursos según la modulación para cada UE y la cantidad de *layers* que esté recibiendo y transmitiendo, se debe también coordinar la asignación de los recursos en todas las *Component Carriers* (CC) que estén utilizadas en cada UE, y también, asignar recursos simultáneos cuando hay *massive MIMO*. Cuando se diferencian servicios en *slices* distintas se superponen el problema de priorizar recursos entre *slices* y el de agendarlos entre los UE de cada *slice*.

Para el objetivo del proyecto se restringe el problema a implementar un *scheduler* simple, asumiendo para el análisis una situación de una sola portadora, SISO (una sola *layer*), sin *slices* ni *beamforming* y con las restricciones actuales de OAI. En el sistema actual se hace una asignación de recursos en base a *subframes*, armando todos los TB correspondientes a cada milisegundo, independientemente del número de *slots* que depende de la numerología utilizada. Esto no implica pérdida de granularidad, solo la asignación anticipada. Tampoco se utilizan múltiples BWP, aunque su implementación está hecha en OAI. Estas restricciones son las

6.2. Descripción de la Estructura de Software del Sistema

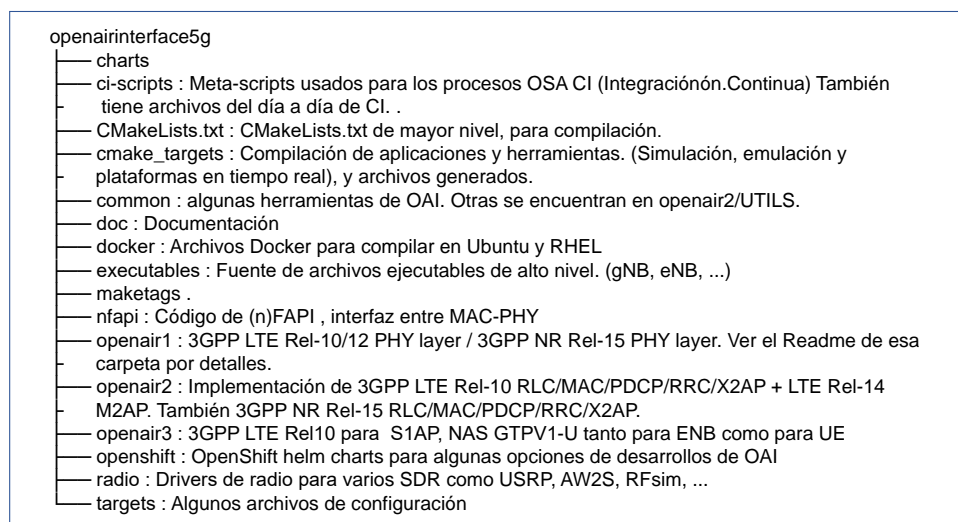


Figura 6.2: Estructura de archivos en git RAN OAI.

mismas definidas en la implementación de la maqueta. Son definiciones que no determinan que el *scheduler* implementado no funcione con varias portadoras, varias *layers* o diferentes BWP en distintos UE. Probablemente sí lo haga, en la medida que el mecanismo de asignación y armado de los TB original se mantiene. Pero se aclara que no fue analizado ni probado en esos escenarios.

6.2. Descripción de la Estructura de Software del Sistema

La maqueta implementada utiliza dos sistemas de software relacionados, pero independientes, el CN y la RAN con el gNB y el UE. En esta descripción se explican las generalidades de la RAN. No se trabajó sobre el código del 5GCN.

El código de la RAN de OAI está desarrollado casi completamente en C, y es un sistema de más de 650.000 líneas de código efectivo. Un millón de líneas totales, sumando comentarios, documentación, archivos de configuración, etc. Son 34.718 archivos que suman 2 GB de datos. Un sistema de esas dimensiones no puede ser estudiado en profundidad en todos sus aspectos en un solo proyecto. Acotamos el estudio y la descripción a las generalidades del funcionamiento a los aspectos vinculados al *scheduler* existente y algunas funciones auxiliares que se utilizaron.

El sistema está compartido en un GitLab de su organización fundadora Eureka y se descarga en la carpeta `openairinterface5g`. La estructura de carpetas del software en su mayor nivel se muestra en la figura 6.2. En la subcarpeta `cmake_targets` están las reglas para compilar y un ejecutable para facilitar esa tarea. Una vez compilado desde una de sus subcarpetas se ejecutan los modems. Los programas `nr_softmodem` (del gNB), `nr_uesoftmodem` (nrUE), `lte_softmodem` (eNB), `lte_uesoftmodem` (UE LTE) y `nr_cuup`, son cinco ejecutables independientes

Capítulo 6. Desarrollo de un Scheduler Alternativo

que se pueden compilar juntos o en forma individual.

Los nodos de radio base se pueden ejecutar como bloque único o también como Central Unit (CU) y Distributed Unit (DU) en programas separados. Hay opciones de compilación que permiten agregar herramientas de prueba y simulación adicionales, como modelos de ruido para simulación de canal, modo de prueba de capa física o telnet server para cambiar configuraciones o parámetros en un sistema funcionando. El driver de SDR que se va a utilizar se debe indicar en las opciones de compilación.

El software va organizado en hilos (threads). Estos se manejan con un sistema de colas propio de OAI. Las capas más bajas que funcionan en tiempo real están asignadas a un hilo adjudicado a un core físico. Se deja primer procesador libre para tareas asíncronas. Los hilos de capas superiores incluyen una interfaz InThread Interface (ITTI) que provee servicios de comunicación asíncrona entre hilos diferentes, accesos a sockets y temporizadores entre otros.

El main general está en el ejecutable `nr-softmodem.c` para el gNB (`openairinterface5g/executables`). Este llama a la función `init_gNB()` que maneja la carga y seteo de configuraciones y el inicio de los hilos de tareas. El hilo general del gNB es el `ru_thread` (ru por Radio Unit). Esta en `openairinterface5g /executables/nr-ru.c`, y tiene el loop permanente sobre el que funciona el conjunto del sistema. En la figura 6.3 se detalla el diagrama de llamados entre las funciones principales que implementan el sistema.

La función `rx_rf()` está en el loop infinito. Se encarga de recibir las muestras IQ del SDR y es llamada una vez por slot. Toma la referencia de tiempo de la estampa de las muestras IQ y lo usa como reloj interno. El conteo de números de trama y slot se deriva de estos *RX samples timestamp*. Siguiendo el diagrama lógico se ve la función `rx_func()` que se encarga de desplegar las tareas globales que manejan la capa física y la capa 2. En la secuencia, la función `NR_UL_indication` llama a la función `gNB_dlsch_ulsch_scheduler`, que es la función global del *scheduler*.

La nomenclatura de variables, parámetros y funciones se basa en las definiciones de la normativa del 3GPP. El comentario más utilizado en el código es la referencia a la norma que los define. El código no está muy comentado y la documentación no es muy didáctica, de modo que esta vinculación directa a las formas y nomenclatura de las normas 3GPP es la mejor ayuda disponible.

Las variables de estado de cada parámetro relevante en cada una de las capas del sistema se guardan en una estructura jerárquica de variables globales. El struct de mayor nivel de esa estructura se llama RC (por *Radio Context*), y está formado por un array de structs de variables para cada una de las capas de protocolos, y un entero por cada capa que indica la cantidad de instancias de esa capa que existe en cada momento. Las funciones de los distintos hilos copian de esa estructura los valores a utilizar, y los salvan en ella al finalizar cada iteración.

El armado de los transport block (TB) a enviar en DL y el desarmado de los recibidos en UL del gNB es la tarea de la capa MAC. Estas dos tareas se dividen en decidir con que prioridad cada usuario accederá a los recursos de transmisión disponibles, a asignar a las señales obligatorias y a los canales de control recursos para ser transmitidos, a reportarle a los usuarios cuándo y en qué frecuencias

6.2. Descripción de la Estructura de Software del Sistema

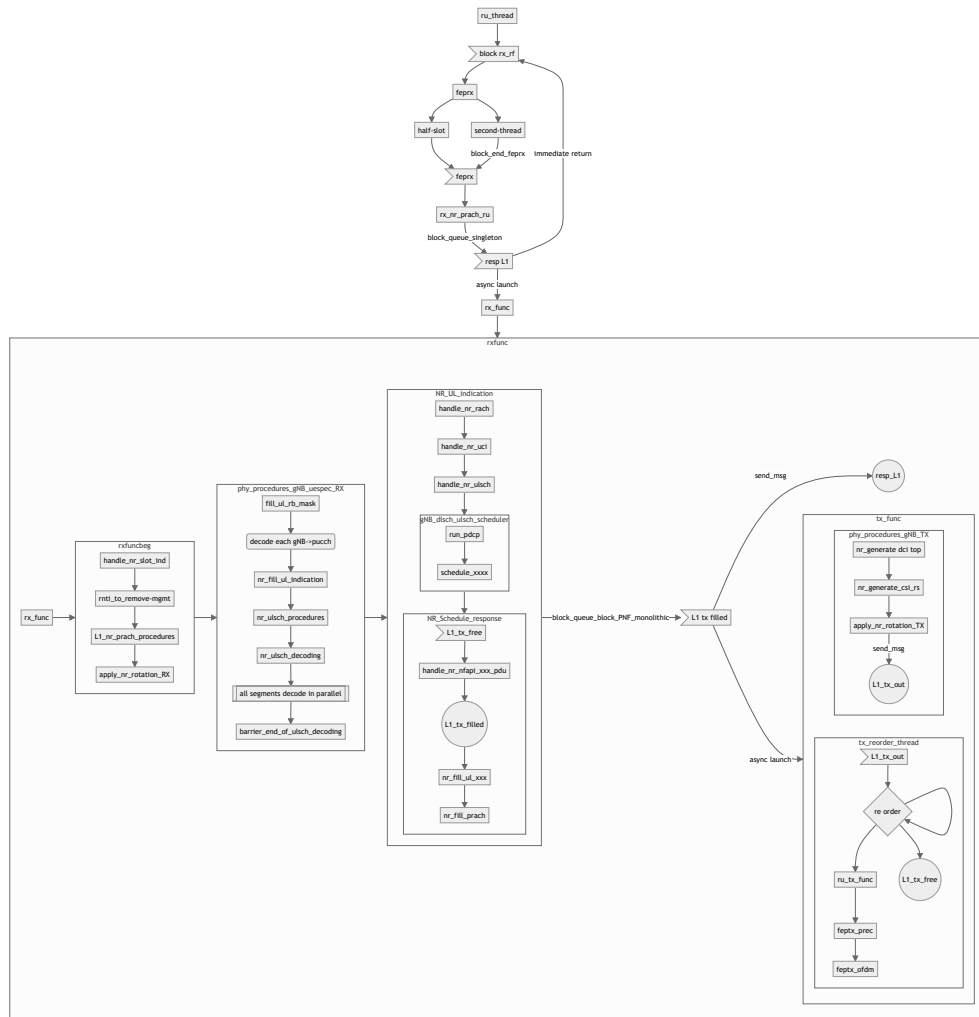


Figura 6.3: Esquema general de hilos de tareas en el sistema OAI-RAN.

escucharán sus mensajes, y finalmente a armar efectivamente la secuencia de bytes a enviar, y posicionarla correctamente.

Los datos a transmitir en Down Link proceden de la capa RLC, donde se encolan en forma independiente los datos de control y de usuario de cada uno de los UE conectados. El *scheduler* decide el orden de multiplexado, y define el armado del *transport block* que será entregado a la capa física, junto con la estructura de la señal a transmitir. Análogamente, la señal de Up Link realiza el proceso inverso, siendo sus *transport blocks* entregados por la capa física a la capa MAC, que los desensambla en flujos de bytes independientes por UE, y por planos de usuario y de control, y separados así se entregan a la capa RLC.

Un proceso igual y opuesto se realiza en el UE, comandado por las instrucciones del *scheduler* del gNB. El gNB le manda mensajes DLICI al UE informándole qué frecuencias y en qué *slots* de tiempo recibirá datos destinados a él. Y recibirá

Capítulo 6. Desarrollo de un Scheduler Alternativo

Gants con anticipación, que le indicarán cuándo tiene agendado transmitir y en qué frecuencias de subportadoras. El teléfono mantiene el grado de libertad de decidir que flujos de datos enviar primero, entre sus requerimientos, en los recursos agendados y puede pedir más recursos si le son insuficientes. También pide recursos cuando tiene nuevos datos para enviar.

Este proceso se realiza cíclicamente. El gNB *scheduler* opera asignando los recursos por *slot*. La cantidad de bytes totales disponibles en cada *slot* es la Transport Block. Su tamaño Transport Block Size (TBS) es variable, según el índice de modulación, la tasa de codificación, el número de portadoras, y el número de *layers*, y los recursos utilizados por funciones de medida y control, con cada UE.

La función gNB_dlsch_ulsch_scheduler lleva un mapa de PRB que reinicia en cada iteración, y se extiende en toda la asignación temporal de esa iteración. Mediante llamados a funciones se generan los datos a transmitir, los copia en la interfaz nFAPI con la capa física, y marca en el mapa de recursos. Inicia con la ubicación del SSB y el RACH cuando corresponde, de las distintas SIB y de los mensajes de control a los UE, con la asignación de reportes CSI y oportunidades de medida SRS. Finalmente llama a las funciones nr_schedule_ulsch y nr_schedule_ue_spec, que hacen el armado de los canales compartidos de downlink y uplink respectivamente. En estas funciones primero se asignan los recursos para los canales de control restantes, luego se ubican los reenvíos de HARQ y finalmente se asignan los demás recursos al primer UE que tenga datos en cola para ser transmitido. Luego al siguiente y así hasta completar los recursos disponibles. Para eso, en cada repetición de la ejecución cíclica, se ordenan los UE conectados según un orden de prioridad definido por el desarrollador.

6.3. Algoritmo Original

En la implementación de OAI, la tarea del scheduler se divide en un preproceso, donde se determina el orden de prioridad, y un post proceso, donde se arma el slot a transmitir con los datos del UE adjudicado y la mensajería de control correspondiente. En la figura 6.4 se muestra el proceso iterativo con que se revisan todos los UE conectados al gNB para seleccionar cuáles y con qué prioridad serán agendados. Los UE con indicación de falla, por ejemplo por exceder el máximo de retransmisiones sin reconocimiento, y los que no tienen datos en cola no entran a la agenda. Los UE que tienen una retransmisión pendiente, tampoco se agendan para nuevos recursos. A ellos se les asignan los mismos recursos anteriores para la retransmisión pendiente antes de la adjudicación de recursos sobrantes.

Los UE se ordenan siguiendo un algoritmo proportional fair. Esto se hace asignando la prioridad en forma directamente proporcional al bitrate que puede alcanzar cada UE en esa subtrama, e inversamente proporcional a los datos transferidos anteriormente. De esta forma se prioriza el uso óptimo del canal, pero a la vez se baja la prioridad a quienes más lo han usado recientemente.

En la práctica, se calcula para cada UE conectado cuál fue su *throughput* en

6.3. Algoritmo Original

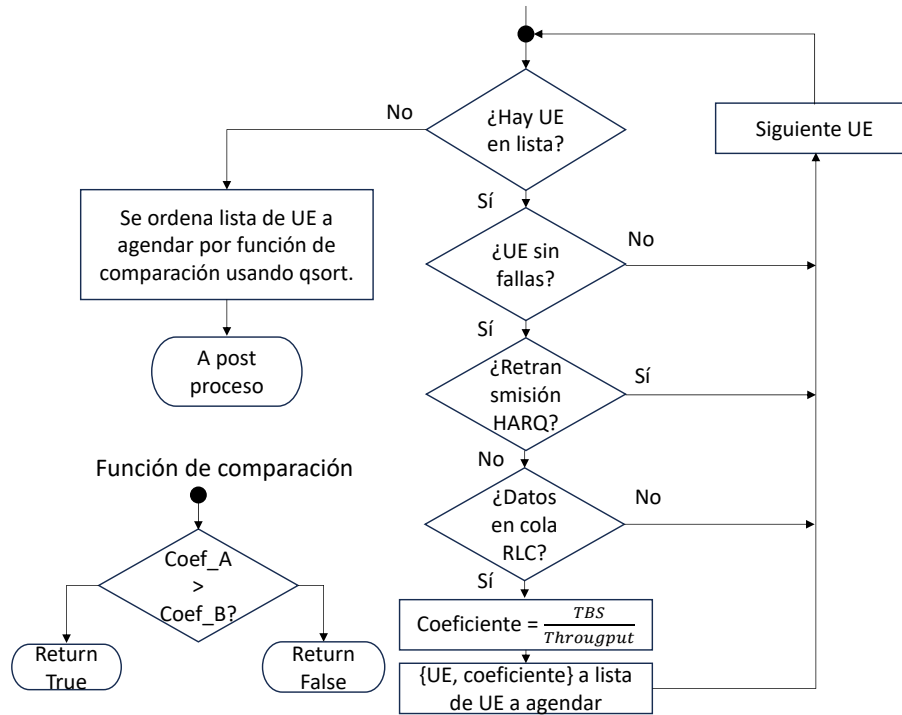


Figura 6.4: Diagrama de flujo de la definición de prioridades de los UE conectados a un gNB en el preproceso del scheduler.

una ventana de los últimos 1000 ms. Se calcula

$$UE \rightarrow dl_thr_ue_n = (1 - a) \times UE \rightarrow dl_thr_ue_{n-1} + a \times b$$

donde a es una fracción de tiempo, $a = 0,0005$, b es la cantidad de bytes enviados a ese UE en el último *slot* y $UE \rightarrow dl_thr_ue$ es la variable donde se actualiza el valor del throughput de cada UE, definida como parte del conjunto de datos de cada UE. Los subíndices n y $n-1$ indican el valor actual y el valor en la iteración anterior de esa variable.

Luego se calcula el tamaño en bytes disponibles de un *transport block* (TB) tipo para ese UE. Se usa una base fija de un PRB durante 20 *slots*, para tener un elemento de comparación independiente de la asignación futura. Los datos de cantidad de portadoras, *layers*, índice de modulación Q y tasa de codificación R utilizada por este UE en ese momento determinan el tamaño del TB base de ese UE en ese *slot*. Es decir, la cantidad máxima de bytes que se le puede enviar en esta oportunidad en un PRB, promediado en los 20 *slots* futuros con las condiciones actuales. El promedio se toma para evitar la injerencia de los símbolos de control, DMRS y slots de UL e híbridos en la decisión. Luego se calcula el coeficiente de prioridad

$$coef_ue = \frac{tbs}{UE \rightarrow dl_thr_ue}$$

Capítulo 6. Desarrollo de un Scheduler Alternativo

que se guarda en un array de datos, que para cada elemento contiene un puntero a la estructura de datos de cada UE en condiciones de ser agendado y el coeficiente calculado. La lista se ordena empezando por el de mayor coeficiente, utilizando una función qsort, y una función de comparación que compara los coeficientes.

En el post proceso, se completan los canales de transporte agregando los datos de control y transfiriendo los datos de usuario a la interfaz de la capa 1. Primero se ubican las retransmisiones, y luego se llenan los datos de usuario, transfiriendo todos los datos posibles del buffer de la capa RLC correspondientes al primer UE de la lista. Si se terminan sus datos, se sigue completando con el siguiente UE. Al repetir el proceso en el siguiente slot, se recalcula la lista con el nuevo orden de prioridad.

6.4. Desarrollo

Para esta implementación se optó por un algoritmo que denominamos Máximo TBS (maxTBS). En él, se le da la mayor prioridad al UE con mayor capacidad de transmisión. Es un algoritmo simple que maximiza la transferencia de la red. No es óptimo desde el punto de vista de los usuarios, pero para el proyecto tiene dos virtudes; es simple, con lo que se mantiene el foco en la implementación y el uso de la herramienta y no en la algoritmia, y es fácil de comprobar su funcionamiento efectivo. Basta con tener condiciones de canal diferentes entre dos UE, y una transferencia de datos masiva en ambos, que requiera de todo el ancho de banda disponible para evidenciar la diferencia de comportamiento entre el sistema original y el sistema modificado.

En la función pf_dl, donde se calcula el coeficiente, se reemplaza el coeficiente calculado por el tamaño del TB para cada UE y el ordenado de la cola de UE para agendar se hace por este valor. El cálculo del TBS se hace según se indica en el inciso 5.1.3.2 de la TS 38.214 para downlink e inciso 6.1.4.2 para uplink. Este proceso se vio en el Capítulo 2. Ya está incluido en el cálculo del coeficiente original, y considerarlo es equivalente a agregarle exponentes de ponderación al algoritmo Proportional Fair, y fijar el exponente del numerador en 1 y el del divisor en 0. Este enfoque representado en la figura 6.5 hace que la modificación se comporte como se busca, pero a la vez sea más flexible, permitiendo explorar otras aproximaciones. La elección de exponentes se hace en tiempo de ejecución, usando el archivo de configuración, como se explica más adelante.

Esta simple modificación funcionaría bien cuando dos UE tienen distinto MCS, pero si ambos tienen igual MCS, entonces la decisión de asignación queda hecha por el orden en la lista de UE en condiciones iguales en la comparación. El primero de la lista, se lleva la mayoría de los recursos. En ese caso, sería esperable que los dos UE tengan asignaciones equivalentes. Para lograr eso, se agregó un entero generado aleatoriamente al struct que tiene los datos del UE y el coeficiente calculado, y se modificó la función de comparación que se le da al algoritmo qsort. Con la nueva comparación mostrada en la figura 6.6, la llamada a la función qsort ordena la lista por coeficiente primero y en caso de empate se ordena por el valor aleatorio. Ese valor aleatorio se recalcula una vez por iteración para cada UE que se ingresa en la

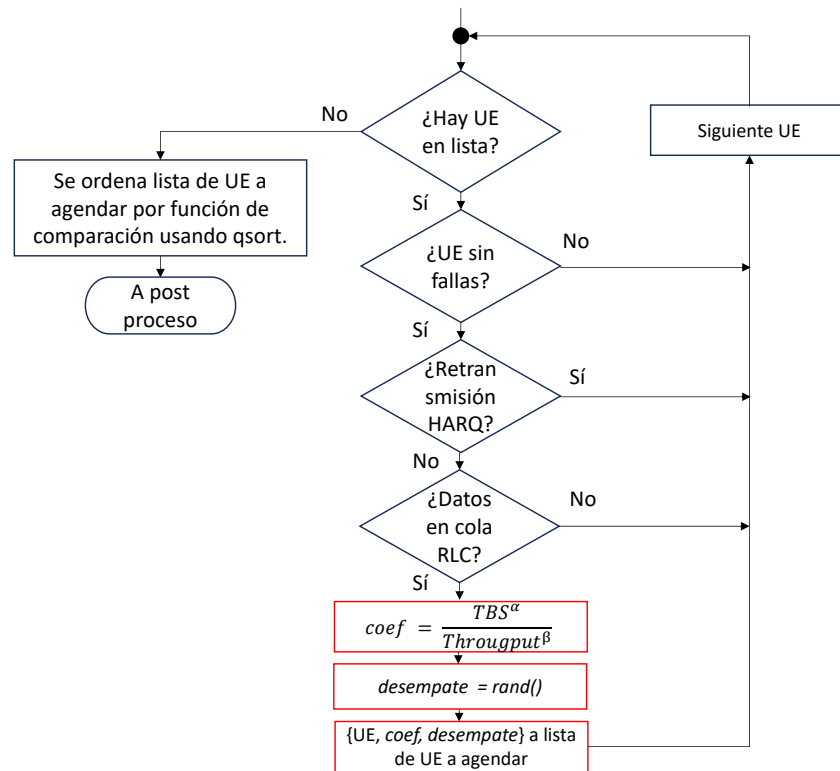


Figura 6.5: Diagrama de flujo de la definición de prioridades de los UE conectados a un gNB en el preproceso del scheduler modificado.

Función de comparación

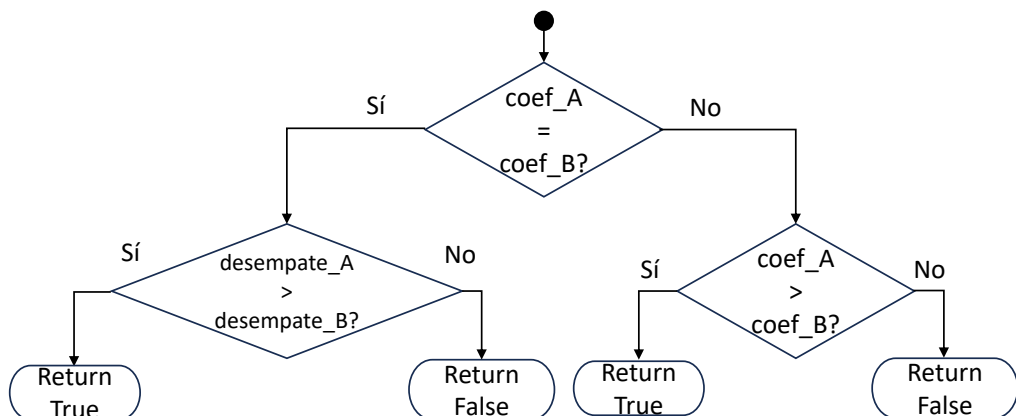


Figura 6.6: Función de comparación para el ordenado de la lista de UE a agendar.

Capítulo 6. Desarrollo de un Scheduler Alternativo

lista a agendar. Con esto se obtiene un reparto equivalente de los recursos, cuando hay igualdad de condiciones. Con este esquema se obtienen tres comportamientos típicos diferentes, con los dos coeficientes en uno, funciona el Proportional Fair original, con un factor de aleatorización en el caso de igualdad de coeficientes, que no tiene impacto, ya que la igualdad de coeficientes en el primer lugar de la lista no se repite en este algoritmo. Si el coeficiente $\alpha = 1$ y $\beta = 0$ tenemos el funcionamiento de MaxTBS ya descrito. Si ambos son cero la distribución es aleatoria, con un comportamiento en media similar a un Round Robin. Otras relaciones pueden estudiarse más adelante.

En el caso MaxTBS a pesar de la prioridad permanente del UE con mayor MCS, siempre hay oportunidad para el segundo, de traficar. Por un lado, debido a que cuando se termina el SDU que la capa RLC envía, los Bytes libres son asignados al siguiente UE de la lista. En 5G, a diferencia de LTE, la capa RLC no concatena paquetes consecutivos de un mismo bearer hasta el tamaño del TB como se hacía en LTE. Solo los particiona si son mayores. El multiplexado lo hace la capa MAC, permitiendo el agregado de nuevos SDU en el mismo slot. Además, cuando un bloque debe ser retransmitido, en la siguiente iteración del scheduler, ese teléfono recibirá recursos para la retransmisión del bloque fallido, pero no entrará en la lista para nuevas asignaciones.

MaxTBS versus MaxCQI

En los primeros intentos se buscó implementar un algoritmo de asignación según el máximo Channel Quality Indication (CQI) que se llamó MaxCQI, con un comportamiento equivalente, pero solo influenciado por el estado del canal. Se asignan recursos al UE que reporta mejor estado del canal. La implementación se logró, pero sólo es utilizable en configuraciones FDD, o con varias *layers*, dado que en TDD, no se utiliza el reporte del CQI para la evaluación del canal. Se considera al canal simétrico, y en el caso SISO, en TDD no se le pide el reporte que incluye el CQI al UE, por lo que no se dispone de ese valor. La selección de la modulación y codificación se hacen con las mediciones de SRS en recepción del gNB y se usa junto con el control de errores de bloque para la selección del MCS de ambas direcciones. Cuando ese algoritmo se probó con una configuración SISO de banda n78, el CQI se mantuvo en 0 (valor para “Out of Range”) y el ordenamiento entre los UE en esas condiciones no depende del estado del canal sino del orden de ingreso a la tabla. En esas pruebas no estaba hecha aún la asignación aleatoria con coeficientes iguales. Probado utilizando dos *layers*, se obtiene el reporte del CQI y se puede probar el algoritmo, pero no es reproducible en el modelo con SDR, por las limitaciones de hardware ya descritas.

Cambiar el criterio de CQI a *Transport Block Size* (TBS) permite resolver ese problema, e incorporar otras dimensiones. Si el estado del canal es bueno y la tasa de errores de bloques es baja el índice de modulación Q_m y el de codificación R , harán mayor al TBS, pero también es mayor si ese UE utiliza una Bandwith Part (BWP) mayor, o si se están usando varias *layers* en ese momento, etc. Todas estas otras dimensiones suman en el sentido de darle más recursos a quien puede usarlos. Por otro lado, el CQI solo representa el estado del canal en DL, se debe utilizar

6.5. Cambio de Parámetros por Configuración

un criterio diferente en UL. En cambio sí se puede usar el criterio de TBS máximo en UL, aunque se debe tener presente que es otro TBS independiente y se debe calcular también para cada tiempo de asignación de recursos.

6.5. Cambio de Parámetros por Configuración

Como se presentó en la sección 4.4, el gNB se configura utilizando un archivo de texto con formato ASN.1 dado por la librería libconfig, utilizada para leerlo. Consiste en una lista de parámetros con sus valores. Algunos parámetros son a su vez listas de parámetros, y de esa forma el archivo toma una estructura jerárquica con valores. Por ejemplo, los valores relacionados a la frecuencia son parte de la lista de configuraciones de gNB y las atenuaciones fijas a agregar en transmisor o receptor son parte de los parámetros del grupo RU.

Los parámetros de configuración se guardan en struct asociados a cada grupo con la forma paramdef.t que se define en archivos como MACRLC_nr_paramdef.h que es el que hemos utilizado aquí. En ese struct se definen variables, se las asocia a los textos que nombran los parámetros en los archivos, se define el tipo del valor esperado, su rango, valor por defecto y algunas posibles banderas, como si es un campo obligatorio, etc. Definiendo un struct de ese tipo se pueden usar los métodos get_config y get_configlist para leer los valores del archivo de configuración y guardarlos en el struct creado. Luego se deben procesar esos datos, tomando las acciones que correspondan a cada configuración.

Para agregar los exponentes que manipularán el *scheduler*, se crearon cuatro variables globales, y se agregaron al formato del struct RC.nrmac que maneja las variables del contexto MAC y RLC. Se agregaron esas variables al tipo de struct MACRLC_nr_paramdef.t que se usa para leer la configuración al inicio del programa, y se copian en las variables globales creadas. En cada iteración del *scheduler*, los exponentes a utilizar se actualizan desde las variables globales de estado del nodo. Esto permite que futuros trabajos se modifiquen en tiempo de ejecución.

En la versión hecha, los parámetros ul_alfa, ul_beta, dl_alfa y dl_beta se pueden definir con valores enteros entre 0 y 4, en la sección MACRLCs del archivo de configuración. El límite de 4 se eligió para poder seguir operando con enteros. Si no se incluyen, el valor por defecto es 1 y el sistema funcionará con el algoritmo *Proportional Fair* original.

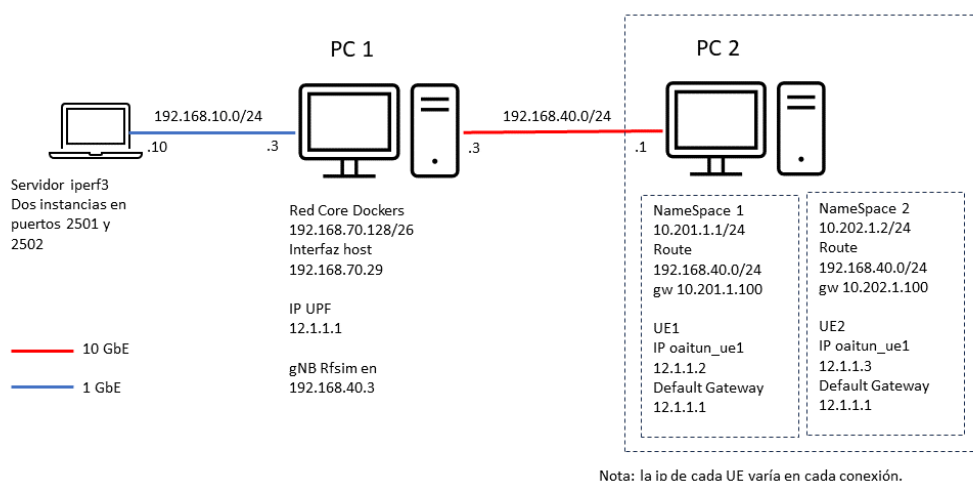
6.6. Ensayos y Resultados

Para probar el funcionamiento del *scheduler* se requiere de escenarios multi-usuarios, y se debe poder incidir sobre las condiciones de funcionamiento de estos. A continuación describimos los escenarios y procedimientos utilizados para probar, y los resultados obtenidos.

Capítulo 6. Desarrollo de un Scheduler Alternativo

Procedimiento de Pruebas con RF Simulator

Para este uso es necesario implementar el caso multiusuario descrito en el manual de instalación. Dado que la aplicación del UE define una interfaz IP en el nodo del usuario, de nombre `oaitun_ue1`, no se puede correr dos instancias de UE en la misma PC porque se genera un conflicto con el nombre de las interfaces. Para correr un segundo usuario es necesario tener otra PC disponible y en red, o un sistema de RAN corriendo en dockers, o en su defecto definir múltiples *network namespaces* y correr un UE en cada uno. Las tres opciones funcionan bien y se probaron. La primera es poco práctica, ya que requiere hardware adicional de PC y de red. La segunda tiene limitaciones porque los programas de softmodem son muy demandantes del hardware. Además, en este escenario donde se modifica el código o la configuración, rearmar el docker cada vez que se cambia algo no es cómodo. El tercer escenario, de la figura 6.7, es el utilizado en las pruebas.



Nota: la ip de cada UE varía en cada conexión.

Figura 6.7: Esquema de maqueta para pruebas de scheduler con RFsim.

En la PC 1 corre el Core Network y un gNB, en banda n78, TDD con 40 MHz de ancho de banda. En la PC 2 corren dos UE en *name spaces* diferentes. La prueba se hace utilizando iperf3 para medir el ancho de banda disponible. El servidor iperf3 se corre en otra laptop que no tiene ruta establecida por la red ip desde la PC2. La PC 2 solo conoce las redes directamente conectadas y los *network namespaces* en ella creados tienen rutas por defecto a través del plano de usuario del *core network*. Con los UE apagados se verifica que no hay conectividad usando ping desde PC2 al servidor iperf3. Se repite la misma verificación desde cada uno de las consolas de los dos *namespaces*. Luego, se corren dos instancias de servidor iperf3 en puertos diferentes, se verifica el ancho de banda de cada UE por separado, y finalmente se hacen dos pruebas simultáneas en ambos UE para comparar resultados. Todo el procedimiento se realiza una vez en UL y otra en DL.

Además de tener varios UE, se requiere variar las condiciones del canal. Para eso se agrega un modelo de ruido AWGN sumado al canal y se ajusta en condiciones

6.6. Ensayos y Resultados

diferentes para cada UE. Eso impone condiciones de canal diferentes, que es la condición requerida para la prueba. En la figura 6.8 se pueden ver los *scopes* de las señales recibidas en dos UE. Se observa claramente la diferencia en las constelaciones por efecto del ruido blanco agregado en uno de ellos. Una limitante de este modelo de prueba, es que el ruido AWGN fue sumado a la entrada de cada UE, afectando solo el DL. No se logró agregar ruido AWGN diferenciado por usuario en la entrada del gNB, por lo que no se pudo usar este escenario para pruebas del *scheduler* en UL.

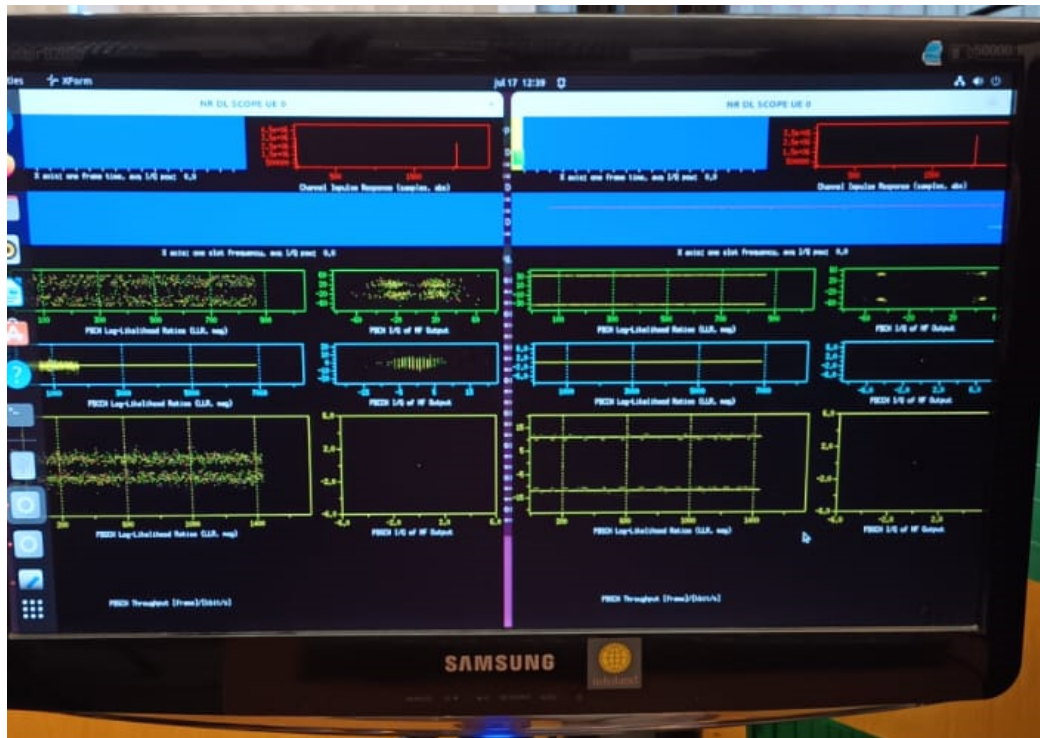


Figura 6.8: Scope de los dos UE, uno con ruido y el otro no.

Hasta el momento de las pruebas, el modelo de canal asociado al RFsim para los nodos de 5G está poco desarrollado, y solo admite el agregado de ruido blanco, sin variar condiciones de enlace, retardo u otros modelos de ruido. Estas funciones están previstas en el desarrollo futuro.

Además de la prueba cuantitativa, se agregó al software la exportación a un archivo de la asignación de PRB realizada a cada UE, para su análisis cualitativo. Se guardan los datos de ubicación temporal (trama y slot) y para cada UE adjudicado en esa iteración, PRB inicial, número de PRBs y símbolos OFDM asignados. Eso permite comparar el detalle del comportamiento micro, entre una asignación y las siguientes, y también el comportamiento estadístico, acumulando los resultados en escalas de tiempo mayores.

Este procedimiento se usa para los dos *scheduler* y se compararan los resultados. Se realizaron series de prueba en condiciones de igualdad del canal y en

Capítulo 6. Desarrollo de un Scheduler Alternativo

condiciones diferentes. En el Apéndice C se listan las pruebas realizadas. Se detalla más adelante un par de pruebas representativas que evalúan el trabajo hecho.

Procedimiento de Pruebas con Interfaz de Radio

Para probar con interfaz de radio se requiere de tres SDR o de un SDR y dos teléfonos. Dado que se dispone de varios SDR Ettus B200, con menor ancho de banda máximo que los X310, se configura un escenario TDD, con 24 PRB y SCS de 30 KHz, utilizando la banda n41. Eso configura un ancho de banda de 10 MHz para esta prueba, de modo que se pueden usar ambos equipos. En estas condiciones se utiliza un X310 como gNB, otro como un UE y un B200 con una laptop para la conexión del segundo UE, también se realizaron pruebas con dos UE B200 y con los tres nodos usando SDR B200. No se observaron otras variaciones en el comportamiento que las resultantes de los cambios en las condiciones de radio.

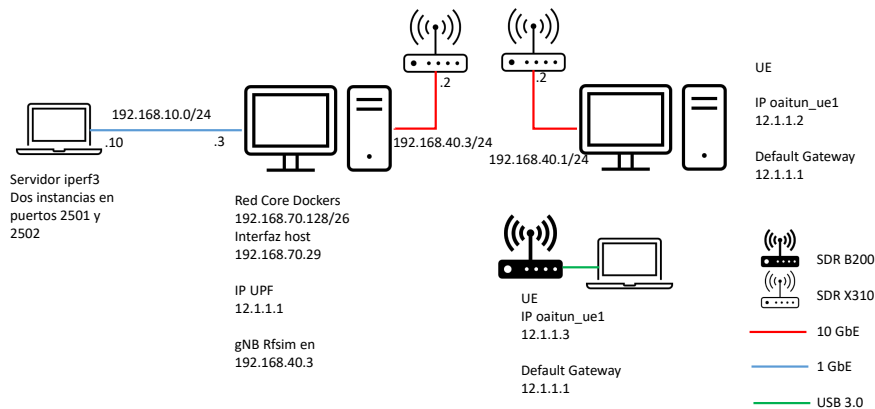


Figura 6.9: Esquema de maqueta para pruebas de scheduler con tres SDR.

Para imponer condiciones de enlace iguales o diferentes en los UE, se mueve uno de ellos, mientras se monitorea el MCS de ambos en el log del gNB. Pero dejarlos quietos no asegura que se mantenga la condición durante el tiempo de la prueba. En la figura 6.9 se representa el esquema de red utilizado en estos ensayos.

Resultados de Pruebas

En primer lugar se muestran los resultados cualitativos. Aquí se ve que el comportamiento de los algoritmos es el esperado. En las figuras 6.10, 6.11 y 6.12 se presenta la acumulación de Bytes asignados por segundo a cada UE, durante una transmisión simultánea hecha con iperf, que lleva el sistema al límite de su capacidad. En la primera se ve como en Proportional Fair, el ajuste en condiciones

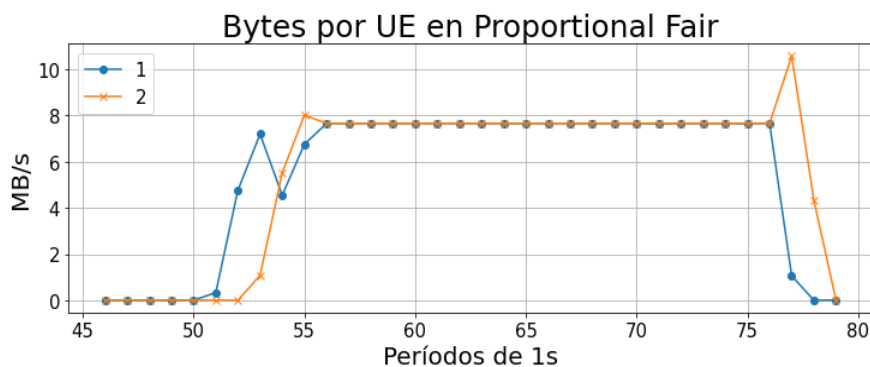


Figura 6.10: Asignación de Bytes en dos UE con Proporcional Fair.

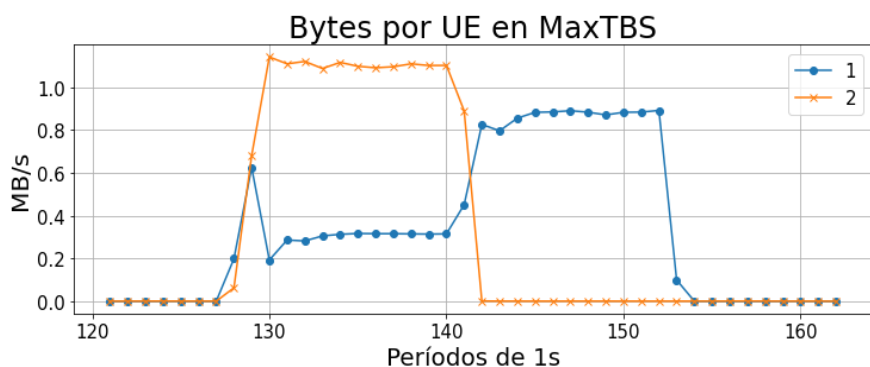


Figura 6.11: Asignación de Bytes en dos UE con MaxTBS.

iguales es casi perfecto, con un transitorio inicial debido a que las pruebas no comienzan en el mismo instante, y otro al final por la misma razón. En la segunda, con algoritmo MaxTBS, el UE con mejor TBS recibe la asignación mayoritaria, y cuando este termina, el segundo puede transmitir a mayor tasa de bits. Finalmente en la tercer captura, dos UE con igual MCS reciben asignaciones aleatorias con resultados promedio similares, con MaxTBS.

En la figura 6.13 se presenta un tramo de la tabla de datos de adjudicación desde el código a un archivo csv. Se ve en las primeras dos columnas el número trama y de slot, que es forman la base de tiempo de las gráficas anteriores. Cada UE se identifica por el valor del Radio Network Temporary Identifier (RNTI), en la tercer columna. Las columnas RB start y RB size indican donde comienza y de cuantos Resource blocks es la asignación para ese UE. En el tramo de planilla de la figura todas las asignaciones son de 106 PRB, lo que indica que se asignan todas las subportadoras a ese UE, ya que esa prueba se hizo en una celda de 40 MHz, 106 PRB. Las columnas de Symbol start y symbol size muestran la asignación

Capítulo 6. Desarrollo de un Scheduler Alternativo

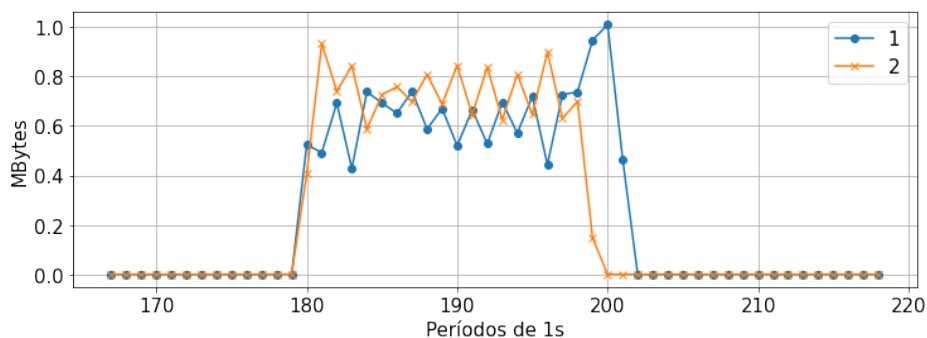


Figura 6.12: Asignación aleatoria de Bytes a dos UE con igual MCS en MaxTBS.

Frame	Slot	RNTI	RB Start	RB size	Symbol start	Symbol Size	DMRS pos	MCS	nrOfLayers	TBS
154	1	17a7	0	106	1	13	804	28	1	10497
154	2	51e8	0	106	1	13	804	28	1	10497
154	3	17a7	0	106	1	13	804	28	1	10497
154	4	51e8	0	106	1	13	804	28	1	10497
154	5	17a7	0	106	1	13	804	28	1	10497
154	6	51e8	0	106	1	13	804	28	1	10497
154	7	17a7	0	106	1	5	4	28	1	3969
154	10	17a7	0	106	1	13	804	28	1	10497
154	11	51e8	0	106	1	13	804	28	1	10497
154	12	17a7	0	106	1	13	804	28	1	10497

Figura 6.13: Alternancia de la asignación en Proportional Fair en régimen. Se observa que en la misma trama se asigna un slot a un UE identificado por su RNTI, y el siguiente al otro.

en el dominio temporal durante cada slot. La columna MCS indica en este caso que se usa el máximo valor de índice de modulación y tasa de codificación, lo que sucede al ser un enlace simulado, y en la columna TBS se ve la cantidad de bytes asignados al UE en ese slot.

Al observar el comportamiento de corto plazo, se ve que las asignaciones hechas por el algoritmo Proportional Fair de la primera gráfica anterior, una vez en régimen, asigna un slot completo a cada UE, alternando entre ellos conforme se cambian los coeficientes. El slot 7, que es híbrido y solo tiene 5 símbolos OFDM de DL, no cambia lo suficiente el coeficiente, y el siguiente slot de DL se vuelve a asignar al mismo UE.

En el ejemplo siguiente (figura 6.14) se ve un tramo de asignación hecha por Max TBS. En él, la mayoría de los slots, que en este caso son de 24 PRB cada uno, se asignan al UE con mayor TBS. El comportamiento es reiterado a lo largo del funcionamiento, se eligió este tramo para mostrar que a pesar de que la diferencia entre ambos TBS es mínima, el comportamiento esperado es muy marcado. Finalmente el caso de la figura 6.15 es otro tramo de la misma prueba, en el que ambos UE tienen el mismo MCS, y las asignaciones no siguen un patrón fijo, sino que responden a una distribución pseudoaleatoria.

El otro análisis interesante surge de comparar cualitativamente el comportamiento del throughput en ambos UE cuando se les exige al máximo en forma simultánea. En la tabla 6.1 se presentan los resultados de una secuencia de pruebas. Se puede ver que con el scheduler original se intenta mantener la equidad del reparto del tráfico, en especial cuando ambos UE están en las mismas condicio-

6.6. Ensayos y Resultados

Frame	Slot	RNTI	RB Start	RB size	Symbol start	Symbol Size	DMRS pos	MCS	nrOfLayers	TBS
791	10	436d	0	24	2	12	12	804	16	1008
791	11	436d	0	24	2	12	12	804	16	1008
791	12	436d	0	24	2	12	12	804	16	1008
791	13	51c5	0	24	2	12	12	804	15	944
791	14	436d	0	24	2	12	12	804	16	1008
791	15	436d	0	24	2	12	12	804	16	1008
791	16	436d	0	24	2	12	12	804	16	1008
791	17	436d	0	24	2	4	4	16	1	325
792	1	436d	0	24	2	12	12	804	16	1008
792	2	436d	0	24	2	12	12	804	16	1008
792	3	436d	0	24	2	12	12	804	16	1008

Figura 6.14: Asignación en Max TBS con distinto MCS. Se puede ver que casi todos los slots son asignados al UE con MCS mayor.

Frame	Slot	RNTI	RB Start	RB size	Symbol start	Symbol Size	DMRS pos	MCS	nrOfLayers	TBS
783	4	436d	0	24	2	4	4	15	1	301
783	5	436d	0	24	2	12	12	804	15	944
783	6	51c5	0	24	2	12	12	804	15	944
783	7	436d	0	24	2	4	4	15	1	301
783	10	436d	0	24	2	12	12	804	15	944
783	11	436d	0	24	2	12	12	804	15	944
783	12	51c5	0	24	2	12	12	804	15	944
783	13	51c5	0	24	2	12	12	804	15	944
783	14	51c5	0	24	2	12	12	804	15	944
783	15	51c5	0	24	2	12	12	804	15	944
783	16	436d	0	24	2	12	12	804	15	944

Figura 6.15: Asignación en Max TBS con igual MCS. La adjudicación de slots es pseudoaleatoria.

nes. Cuando uno de ellos tiene ruido de canal agregado en recepción, su capacidad de tráfico es menor, pero al compartir recursos, se sacrifica más el que tiene más capacidad intentando mantener el tráfico en ambos UE.

Cuando se observan los resultados en el scheduler modificado, se puede ver que hay una diferencia importante en el comportamiento, pero en el caso de igualdad de condiciones de canal, se reparten en órdenes similares. Dado que la capacidad del canal es mayor a los resultados de transferencias obtenido, es razonable que el UE con menos recursos, mantenga una cantidad de tráfico en curso. Cuando uno de los UE tiene ruido agregado, es quien se queda con menos recursos en todos los casos.

Las pruebas de la tabla muestran el resultado total de la combinación de la asignación de recursos, con el manejo de errores, y de la capacidad del sistema en su hardware y su software de manejar los flujos de datos utilizados. Como es de interés mostrar la asignación en condiciones límite, se lleva al límite también las demás condicionantes. Se observan en los logs de los UE que aumentan las retransmisiones en forma importante, y en los logs de UE y de gNB se observan con frecuencias mensajes de error, con pérdidas de SDU por excederse en el tiempo previsto. Si la prueba no se hace al máximo de la capacidad, se observan tiempos muertos en la transmisión, por lo que la asignación de recursos en tiempos cortos es equivalente a tiempos de transmisiones independientes de un UE y de otro. A modo de ejemplo, se hace una prueba con dos ping simultáneos con paquetes de 1000 bytes cada uno desde dos UE. Se ve en la figura 6.16 que solo algunos slots tienen asignaciones.

Al realizar el mismo conjunto de pruebas con enlace de radio reales se obtienen

Capítulo 6. Desarrollo de un Scheduler Alternativo

	Proportional Fair				MaxTBS			
	UE1 DL Mb/s	UE1 UL Mb/s	UE2 DL Mb/s	UE2 UL Mb/s	UE1 DL Mb/s	UE1 UL Mb/s	UE2 DL Mb/s	UE2 UL Mb/s
2 UE con SDR	141	18.7			138	18.2		
UE 1 con tráfico			149	15.9			140	15.7
UE 2 con tráfico	86.8	9.15	84.0	8.78	57.9	4.2	112	15.7
UE 1 y UE 2								
UE 2 +AWGN								
UE 1 con tráfico	77.9	16			74.0	10.6		
UE 2 con tráfico			23	15.8			29.5	9.8
UE 1 y UE 2	51.5	8.5	16.1	8.2	78.3	8.9	12.1	2.9

Tabla 6.1: Tabla de performances medidas con dos UE usando RFSim, en una celda de 40 MHz de ancho de banda (106 PRB). Se comparan los resultados obtenidos con scheduler Proportional Fair y MaxTBS.

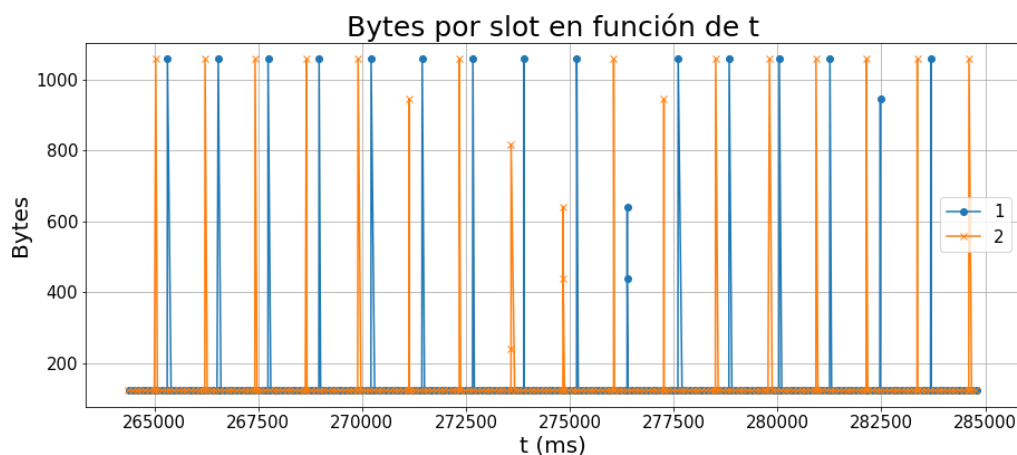


Figura 6.16: Asignación a dos UE durante pruebas de ping con 1000 Bytes. Se puede ver que casi todos los slots quedan sin asignarse. Es indiferente el scheduler utilizado.

comportamientos semejantes. El resultado se presenta en la tabla 6.2. En este caso, con enlace de 10 MHz y MCS menores por las condiciones de enlace, los throughputs son todos menores. En cambio las proporciones entre ellos se mantienen semejantes al caso de RFSim. En el primer juego, dos UE con condiciones de radio semejantes pero no idénticas, obtienen un reparto equitativo en Proportional Fair y desbalanceado en MaxTBS. Alejando el UE con mejor comportamiento en MaxTBS se logra invertir la asignación.

Estas pruebas hechas con enlaces funcionando tienen la dificultad de que las condiciones de enlace son variables, la cantidad de errores, y en consecuencia el MCS de cada UE en cada instancia no es seleccionable, ni se mantiene constante.

	Proportional Fair				MaxTBS			
	UE1 DL Mb/s	UE1 UL Mb/s	UE2 DL Mb/s	UE2 UL Mb/s	UE1 DL Mb/s	UE1 UL Mb/s	UE2 DL Mb/s	UE2 UL Mb/s
2 UE con SDR								
UE 1 con tráfico	8.82	1.64			5.96	1.32		
UE 2 con tráfico			7.02	1.01			6.96	1.17
UE 1 y UE 2	5.36	1.43	4.45	0.75	1.90	0.86	6.69	0.86
UE2 alejado								
UE 1 con tráfico					8.15	2.98		
UE 2 con tráfico							6.36	0.77
UE 1 y UE 2					7.88	2.98	2.35	0.48

Tabla 6.2: Pruebas de performance con scheduler Proportional Fair y MaxTBS utilizando SDR, en una celda de 10 MHz (24 PRB).

Esto hace que las pruebas tengan resultados con variaciones importantes, aún cuando el patrón de comportamiento es el esperado. Las pruebas que se presentan aquí son representativas, pero no tienen valor estadístico. Son solo casos aislados de pruebas típicas. Es por eso, que los resultados que se presentan en las tablas cualitativas, solo sirven de comparación indicativa, ya que todos los valores de la tabla corresponden a instancias similares pero independientes, salvo los resultados de dos UE medidos simultáneamente. Un comportamiento equivalente a todo el descrito en este capítulo se observa en Up Link, donde se implementó el mismo cambio.

6.7. Resumen

Se ha investigado la la arquitectura del software de la RAN OAI, la forma de trabajo del *scheduler* existente y su inserción en la implementación del stack de protocolos, y las funciones de inicialización y configuración del sistema. Se logró implementar una alternativa funcional con comportamiento diferente en el sentido buscado. El *Scheduler* Max TBS prioriza al UE con mejores condiciones para la transmisión, aunque mantiene con datos transmitidos al siguiente UE en las ventanas de oportunidad que se presentan.

Como resultado del trabajo, quedó implementada una función con una interfaz externa que permite variar los coeficientes de ordenado de los UE alterando los exponentes de la relación del coeficiente por configuración al correr cada instancia y puede evolucionar a un control dinámico, con el sistema funcionando. Es un primer paso dado en el camino de utilizar esta maqueta en las investigaciones vinculadas a la optimización de la asignación de recursos existente en la Facultad.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 7

Conclusiones

A lo largo de este trabajo, se logró seleccionar, instalar y probar un sistema 5G SA completo, capaz de dar servicio por interfaz de radio a usuarios con dispositivos del mismo sistema y también a teléfonos comerciales. Los usuarios son capaces de intercambiar datos entre sí, y de conectarse a redes externas, navegando en internet, utilizando el acceso provisto por la red de maqueta.

Esta maqueta se realizó con el objetivo de proporcionar una herramienta de pruebas que permita a los grupos de investigación de facultad probar sus desarrollos en un entorno comparable a una red real.

7.1. Cumplimiento de Objetivos

El objetivo de este proyecto es implementar una maqueta básica de una red 5G Stand Alone, de emulación, que incluya las funciones de Core Network (CN), de Base Station (BS) y de User Equipment (UE), utilizando herramientas existentes, desarrolladas en código abierto, e implementando la interfaz de radio con Software Defined Radio (SDR).

En el plan de proyecto se establecieron como criterios de éxito el cumplimiento de las siguientes metas, que evaluamos a continuación.

- El criterio de éxito mínimo es establecer la conexión extremo a extremo, permitiendo realizar una descarga en downlink y otra en uplink a una tasa mínima de 10 Mbps. Se cumplió con el objetivo, como se describió en los capítulos 4 y 5.
- El segundo criterio de éxito es evaluar la tasa de transferencia máxima del sistema implementado en UL y DL y compararla con los valores teóricos esperables. Se realizó en el capítulo 5.
- El tercer criterio es la entrega de manuales de instalación, configuración, y procedimientos de operación del sistema implementado. Se incluyen en el Apéndice A.

Capítulo 7. Conclusiones

- El cuarto criterio, es lograr implementar un scheduler alternativo funcional. Esto fue presentado en el capítulo 6.
- Un criterio de éxito adicional, y opcional, fue conectar un celular comercial a la red de maqueta, lo que se logró y presentó en el capítulo 4.

7.2. Caracterización de la Maqueta

Se logró caracterizar el funcionamiento de la maqueta en ambientes tales como radio frecuencia en aire (aire-RF), radio frecuencia en cables coaxiales con atenuadores (coaxial-RF) y en un ambiente simulado con RF Simulator (RFsim). En aire-RF y coaxial-RF si bien los resultados en cuanto a *throughput* están alejados de los establecido en la teoría, sirven de marco para futuros estudios en cuanto a la diversidad de hardware utilizable (SDR, servidores, dispositivos móviles comerciales, etc), además de profundizar en técnicas para ajustar parámetros de configuración de los enlaces. Estos parámetros están accesibles en OAI tanto para el gNB como para los UE a través de archivos de configuración o comandos en línea. Es crucial entender cómo interactúan estas entidades, para el correcto balance de los enlaces y evitar saturación o potencias débiles en recepción que contribuyan a un bajo rendimiento. En este sentido el mayor aporte de este proyecto se centra en funcionamiento y los manuales que sirvan de base para futuros estudios.

Por otro lado RFsim ha sido una herramienta importante para el estudio de la plataforma en general, y en particular al observar la interacción de los protocolos en todos los segmentos de la red OAI, core y red de acceso. No es útil para caracterizar el *throughput* ya que los resultados obtenidos en el capítulo 5, no concuerdan con lo establecido en la teoría. De todas formas es una herramienta fundamental e introductoria en el estudio de OAI, que permite una base de conocimiento, para luego escalar hacia instalaciones con hardware de radio frecuencia.

7.3. Los Próximos Pasos

En nuestra opinión es conveniente dar continuidad al trabajo de estudio y desarrollo de la maqueta implementada. Una versión todo en uno en Dockers es fácilmente replicable y útil para ejercicios de laboratorio en materias vinculadas al tema. Experimentar con configuraciones MIMO 2x2 usando dos UBX-160 en un X310 y dos B200 en el mismo UE, es factible. También es posible a partir de las versiones actuales, agregar varios *slices*, manejar múltiples *bandwidth parts*, probar conexiones directas de UE a UE (*Device to Device*) o la señal de posicionamiento PRS, todas funciones disponibles.

Pero de mayor interés nos parece explorar las configuraciones particionando CU y DU, generando una conexión de transporte intermedia. Agregar la interfaz E2, que reporta el estado de todas las variables internas de los nodos a un elemento de control inteligente externo. Parece de sumo interés para los proyectos de investigación actuales, experimentar con la instalación del RAN Intelligent Controller FlexiRAN, y del orquestador en el que están trabajando en el proyecto Mosaic5G.

7.3. Los Próximos Pasos

Finalmente nos parece importante encontrar una ubicación física semipermanente, donde se pueda realizar una instalación de sistema radiante, que habilite dejar templates con el balance de potencia adecuado, que facilite el uso en todos los casos que no aborden la temática específica de RF. Y recomendamos que se comple un reloj con GPS para completar el equipamiento.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice A

Manuales

En esta sección se recopilan las hojas de datos del hardware, luego se escribe un manual con los procedimientos de instalación de los sistemas, y un manual de usuario donde se listan procedimientos de configuración y operación utilizados en el proyecto.

A.1. Datos del Hardware

A.1.1. USRP X310

Funcionalidades:

- Dos posiciones para conectar tarjetas de radiofrecuencia de banda ancha de hasta 160 MHz cada una (tarjetas UBX, CBX, WBX, SBX).
- Las tarjetas de RF cubren desde corriente continua hasta 6 GHz
- FPGA Xilinx Kintex-7 para DSP de alto rendimiento (XC7K410T)
- 10 Gigabit Ethernet dual – 2x RX a 200 MSPs por canal
- 10 Gigabit Ethernet dual – 4x RX a 80 MSPs por canal
- PCIe Express (para PC de escritorio) – 200 MS/s Full Duplex
- ExpressCard (para Laptop) – 50 MS/s Full Duplex
- Dual 1 Gigabit Ethernet – 25 MS/s Full Duplex
- Arquitectura de reloj flexible
- Tasa de muestreo configurable.
- OCXO disciplinado por GPS opcional
- Operación coherente utilizando los distribuidores de sincronismo OctoClock y OctoClock-G



Figura A.1: USRP X310.

- Gabinete compacto y robusto, de medio rack de ancho y 1U de altura.
- I/O digital accesible en el panel frontal para personalizar la FPGA

La arquitectura de UHD provee compatibilidad con

- GNU Radio
- C++/Python API
- Amarisoft LTE 100
- OpenBTS, OAI y srsRAN
- Otros software y plataformas de terceros.

A.1.2. Daughterboard UBX 160

Funcionalidades

- Amplio rango de frecuencias: 10 MHz – 6 GHz
- Ancho de banda: llega hasta 160 MHz



Figura A.2: UBX 160.

- Compatible con la serie X de USRP
- RF shielding
- Operación full duplex con frecuencias de TX y RX
- Sincronización de sintetizador para aplicaciones que lo requieren
- Operación coherente o alineada en fases

A.1.3. LP0965 Antenna

LP09650 antena direccional Log-periódica de 850 MHz a 6.5 GHz hecha en PCB, de 5-6dBi de ganancia. Trabaja con cualquier tarjeta de RF que opere en ese rango.

A.1.4. Tarjeta Ethernet 10 Gbps X520-DA2

Especificaciones:

- Port Configuration Dual Cable Medium Copper
- Cable para conexión directa con conectores tipo SFP+ hasta 10 m
- Tasa de datos por puerto 10/1GbE
- Intel® Virtualization Technology for Connectivity (VT-c) Yes
- Speed & Slot Width 5.0 GT/s, x8 Lane
- Controller Intel® 82599 10 Gigabit Ethernet Controller



Figura A.3: LP0965 Antenna.



Figura A.4: Tarjeta 10 Gbps.

A.2. Instalación de la plataforma



Figura A.5: Cable SFP.

A.1.5. Servidores

Se utilizan 2 servidores con siguientes características, cuyas especificaciones técnicas son:

- Procesador 12th Gen Intel Core™i7-12700
12 Cores Intel®Integrated Graphics Frecuencia: 2.1- 4.9 GHZ Cache Size: 35 MB Threads: 20
- Placa Madre, con socket para el anterior procesador, con puertos vga, hdmi, usb y usbc
- RAM: 16 GB, DDR5
- Almacenamiento: 512 GB SSD
- Gabinete y fuente de poder acorde a las especificaciones anteriores
- Monitor, mínimo 17 pulgadas, vga o hdmi
- Recursos gráficos proporcionados por el procesador y la placa madre. No se requiere tarjeta gráfica especial.
- Sistema operativo Ubuntu 22.04.1 LTS, Kernel 6.2.0-33-generic.

A.2. Instalación de la plataforma

Se implementa a través del tutorial de instalación de la documentación OAI, link de la fecha 12/02/2023: [doc/NR SA CN5G gNB USRP COTS UE Tutorial](#)

Apéndice A. Manuales

Driver para USRP

Procedimiento para la instalación de drivers para el USRP.

```
sudo apt install -y libboost-all-dev libusb-1.0-0-dev doxygen
python3-docutils python3-mako python3-numpy python3-requests
python3-ruamel.yaml python3-setuptools cmake build-essential

git clone https://github.com/EttusResearch/uhd.git ~/uhd
cd ~/uhd
git checkout v4.5.0.0
cd host
mkdir build
cd build
cmake ../
make -j $(nproc)
make test # This step is optional
sudo make install
sudo ldconfig
sudo uhd_images_downloader
```

A.2.1. 5GCN

Se describe paso a paso el proceso para la instalación y puesta en funcionamiento del Core Network OAI 5G.

PRERREQUISITOS:

En los prerrequisitos se instala la herramienta de de acceso remoto "putty", el paquete de transporte apt-transport-https para descargas utilizando HTTPS, el paquete de Certificate Authority para los certificados usados en HTTPS, el comando curl (client url) usado para descargas compatible con la mayoría de los protocolos, el paquete software-properties-common, para manejar actualizaciones y desinstalaciones en la distribución y el software de terceros.

Finalmente se instala la aplicación docker y la herramienta docker-compose, que permite configurar todos los contenedores en un único archivo YAML, y correrlos o pararlos todos juntos en un único comando.

```
sudo apt install -y git net-tools putty
sudo apt install -y apt-transport-https ca-certificates curl
software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |sudo apt-key add-

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/
linux/ubuntu (ls\_realease -cs) stable"
sudo apt install -y docker docker-ce
sudo usermod -a -G docker whoami
```

A.2. Instalación de la plataforma

```
sudo curl -L "https://github.com/docker/compose/releases
/download/v2.12.1/docker-compose -s -m" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

CORE NETWORK SET UP:

Luego se descargan los dockers correspondientes a cada una de las funcionalidades del Core 5G, cada una de ellas en un docker exclusivo.

```
git clone https://gitlab.eurocom.fr/oai/cn5g/oai-cn5g-fed-git
~/oai-cn5g

docker pull oaisoftwarealliance/oai-amf:develop
docker pull oaisoftwarealliance/oai-nrf:develop
docker pull oaisoftwarealliance/oai-smf:develop
docker pull oaisoftwarealliance/oai-udr:develop
docker pull oaisoftwarealliance/oai-udm:develop
docker pull oaisoftwarealliance/oai-ausf:develop
docker pull oaisoftwarealliance/oai-spgwu:develop
docker pull oaisoftwarealliance/trf-gen-cn5g:latest
docker image tag oaisoftwarealliance/oai-amf:develop oai-amf:develop
docker image tag oaisoftwarealliance/oai-nrf:develop oai-nrf:develop
docker image tag oaisoftwarealliance/oai-smf:develop oai-smf:develop
docker image tag oaisoftwarealliance/oai-udr:develop oai-udr:develop
docker image tag oaisoftwarealliance/oai-udm:develop oai-udm:develop
docker image tag oaisoftwarealliance/oai-ausf:develop oai-ausf:develop
docker image tag oaisoftwarealliance/oai-spgwu:develop oai-spgwu:develop
docker image tag oaisoftwarealliance/trf-gen-cn5g:latest trf-gen-cn5g:
latest
```

ARCHIVOS DE CONFIGURACION OAI CN5G: Cómo último paso del CORE se descargan y se instalan los archivos de configuración del Core Network y la base de datos.

El archivo de configuración se copia en /oai-cn5g-fed/docker-compose. El archivo de base de datos se copia en /oai-cn5g-fed/docker-compose/database.

```
wget -O~/oai-cn5g-fed/docker-compose/docker-compose-basic-nrf.yaml
wget -O~/oai-cn5g-fed/docker-compose/database/oai_db.sql
```

SIM CARD:

Para utilizar teléfonos comerciales COTS, se debe grabar una SIM libre, utilizando el software uicc-v2.6. del Open Cells Project con el siguiente comando:

```
sudo ./program_uicc --adm 12345678 --imsi 000000001 --acc 0001 --key
fec86ba6eb707ed08905757b1bb44b8f --opc\ C42449363BBAD02B66D116BC975D77CC1
-spn "OpenAirinterface" --authenticate
```

Apéndice A. Manuales

Las claves y el IMSI utilizados aquí están incluidos en la base de datos por defecto. Las tarjetas SIM traídas para el proyecto fueron pregrabadas por el Open Cells Project, pero se adquirió la grabadora, y se pueden reconfigurar.

A.2.2. gNB

La siguiente es un ejemplo de secuencia de comandos para la instalación y ejecución de OAI gNB, en el sistema operativo Ubuntu 22.04.

```
#Prerrequisitos:
```

```
sudo apt install -y libboost-all-dev libusb-1.0-0-dev doxygen python3-  
docutils python3-mako python3-numpy python3-requests python3-ruamel.yaml  
python3setuptools cmake build-essential
```

```
git clone https://github.com/EttusResearch/uhd,git ~/uhd
```

```
git checkout v4.3.0.0
```

```
cd host
```

```
mkdir build
```

```
cd build
```

```
cmake ../
```

```
make test
```

```
sudo make install
```

```
sudo ldconfig
```

```
sudo uhdImagesDownloader
```

```
# Obtener código fuente openairinterface5g:
```

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

```
cd ~/openairinterface5g
```

```
git checkout develop
```

```
# Instalar dependencias de OAI:
```

```
cd ~/openairinterface5g
```

```
source oaienv
```

A.2. Instalación de la plataforma

```
cd cmake_targets
./build_oai -w USRP --ninja --nrUE --gNB --build-lib all -c
```

A.2.3. UE

Secuencia de comandos para la ejecución del UE:

```
# nrUE EJECUCION:

source oaienv

cd make_targets/ran_build/build

sudo ./nr-uesoftmodem -r 106 --numerology 1 --band 78 -C 3619200000
--ue-fo-compensation --sa -E --uicc0.imsi 0010100000000001
--uicc0.nssai_sd 1
```

A.2.4. Ejecutar la Plataforma Completa

Ejemplo de ejecución de la plataforma completa, en este caso utilizando URSP X310 como gNB en un PC junto con el CORE y en otra pc un USRP X310, donde corre el UE.

```
#Run OAI CN5G:

cd ~/oai-cn5g-fed/docker-compose

python3 core-network.py --type start-basic --scenario 1

#Run OAI gNB:

cd ~/openairinterface5g

source oaienv

cd cmake_targets/ran_build/build

sudo ./nr-softmodem -O ../ ../ ../targets/PROJECTS/
GENERIC-NR5GC/CONF/gnb.sa.band777.fr1.273PRB.2X2.USRPN300.conf
--sa --usrp-tx-thread-config1 --continuous -tx

#Testear con OAI nrUE:

cd ~/openairinterface5g
source oaienv
```

Apéndice A. Manuales

```
cd cmake_targets/ran_build/build
```

```
sudo ./nr-uesoftmodem -r 106 --numerology 1 --band 78 -C 3619200000  
--ue-fo-compensation --sa -E --uicc0.imsi 001010000000001  
-uicc0.nssai-sd 1
```

A.2.5. RF Simulator

RF Simulator (RFSim) constituye un set de herramientas que permite principalmente, simular los procesos asociados a las capas físicas, y en nuestro casos transmitirlos a través de la interface LAN Ethernet de los servidores.

La operativa consiste en enviar recibir muestras de las señales en fase y cuadratura (I/Q) en el dominio del tiempo, entre los gNBs y los UEs. Todos las opciones y modos de OAI estan disponibles en RFSim y por lo tanto pueden simularse, evaluarse y debuggearse.

Principalemente dos modos de funcionamientos:

1. modo "noS1", el tráfico IP generado es muestreado y enviado a través de una interface de tunel creada entre el gNB y el UE, de nombre *oaitun*. Es utilizada principalmente para aplicaciones de ping e inperf.
2. "phy-test", se genera tráfico de subida (UL) y de bajada (DL), en forma randómica en cada oportunidad de scheduling.

RFSim esta implementado en todos los dispositivos de OAI, y se compilan con los correspondientes complidadores de UE y gNB con la opcion:

```
-w SIMU
```

Para ejecutarlo, hay que agregar al comando la opción:

```
- - rfsim
```

RFSim opciones en archivo de configuración:

```
serveraddr = "server";
```

Nombre del servidor o su direccion IP

```
serverport = "4043";
```

Puerto TCP donde escucha el servidor.

```
options = (); #("saviq"); or/and chanmod"
```

Habilita en canal de simulación.

```
modelname = "AWGN";
```

A.2. Instalación de la plataforma

Opciones de modelado de canal, en este caso con el agregado de ruido blanco gaussiano.

```
IQfile = /tmp/rfsimulator.iqs”;
```

Puede usar capturas de prueba de campo o simulación grabada.

Con la siguiente secuencia de comandos ejecutamos el gNB en modo RFSimulator, asumiendo que el core esta corriendo en otro servidor y que tiene conexión LAN Ethernet con la PC donde corre el gNB

```
#gNB CON RFSimulator:

cd ~/openairinterface5g
source oaienv
cd cmake_targets/ran_build/build

sudo ./nr-softmodem -O ../../../../targets/PROJECTS/GENERIC-NR-5GC/
CONF/gnb.sa.band78.fr1.106PRB.usrpb210.conf
--gNBs.[0].min_rxtxttime 6 --rfsim --sa
```

En la misma PC donde corre el gNB ejecutamos el UE con al siguiente secuencia de comandos:

```
# UE CON RFSimulator:

cd ~/openairinterface5g
source oaienv
cd cmake_targets/ran_build/build

sudo RFSIMULATOR=192.168.10.3 ./nr-uesoftmodem -r 106
--numerology 1 --band 78 -C 3619200000 --rfsim --sa --uicc0.imsi
001010000000001 --uicc0.nssai_sd 1 -d
```

Hasta aquí el funcionamiento es con un servidor donde corre el core 5G y otro servidor o PC con un gNB y un UE en en modo RFSimulator. La interfaz LAN ethernet del PC, simula el canal de radio y a través de el se implementa, por ejemplo iperf para validar performance de la comunicación. Esta configuración puede ser más que suficiente para muchas pruebas de validación y desempeño de la plataforma, pero sin duda hay determinadas funciones que solo pueden validarse agregando más UE. Por ejemplo para validar los algoritmos de asignación de recursos, entre otras. Para poder agregar más UE a el RFSimulator, se debe implementar un espacio de nombres para distinguirlos adecuadamente, el procedimiento se detalla a continuación.

A.2.6. 2 UE con *Network Name Spaces*

Debido a que todas las instancias de UE abren una interfaz virtual con la máquina host que siempre tiene el mismo nombre, no es posible correrlas directamente en una sola máquina.

Para superar esa dificultad, se pueden crear varios network name spaces en la máquina, para correr una instancia de UE en cada uno de ellos. Un network name space es una entidad que funciona como un nodo de red independiente dentro de la PC. Se crean interfaces IP entre el name space y la máquina host, por lo que las interfaces del mismo nombre, en distintos name spaces, son interfaces diferentes sin conflictos. En lo siguiente se detalla como implementarlos, y las rutas IP a agregar para que el nuevo sistema funcione.

Se provee un script con los comandos para crearlos, el cual se distribuye en el gitlab del proyecto (<https://gitlab.fing.edu.uy/maq5g-pfc/maqueta-5g/-/blob/main/doc/multi-ue-rfsim.sh>). Al final de la sección, se copia el contenido del script, para quien quiera experimentar variantes, o no tenga acceso al git.

```
# Crear espacio de nombre:

# El archivo multi-ue-rfsim.sh se debe guardar en la ruta donde
# está nr-uesoftmodem:

root@maq5g:/home/maq5g/openairinterface5g/cmake_targets/ran_build/
build

# Se ejecuta el archivo multi-ue-rfsim.sh con el comando:

sudo ./multi-ue-rfsim.sh -c1 -e

# agregando c2, c3...etc se pueden crear varios UEs

# Con el siguiente comando se puede ver la lista de ues creados:

./multi-ue-rfsim.sh -l

# Se debe agregar el ruteo de ip, en cada ue creado con
# los comandos:

sudo sysctl net.ipv4.conf.all.forwarding=1

sudo iptables -P FORWARD ACCEPT

# Se debe agregar la ruta estática desde el UE al gNB, es la torre
# física donde corre el gNB

sudo route add -net 192.168.10.0/24 gw 10.201.1.100
```


A.2. Instalación de la plataforma

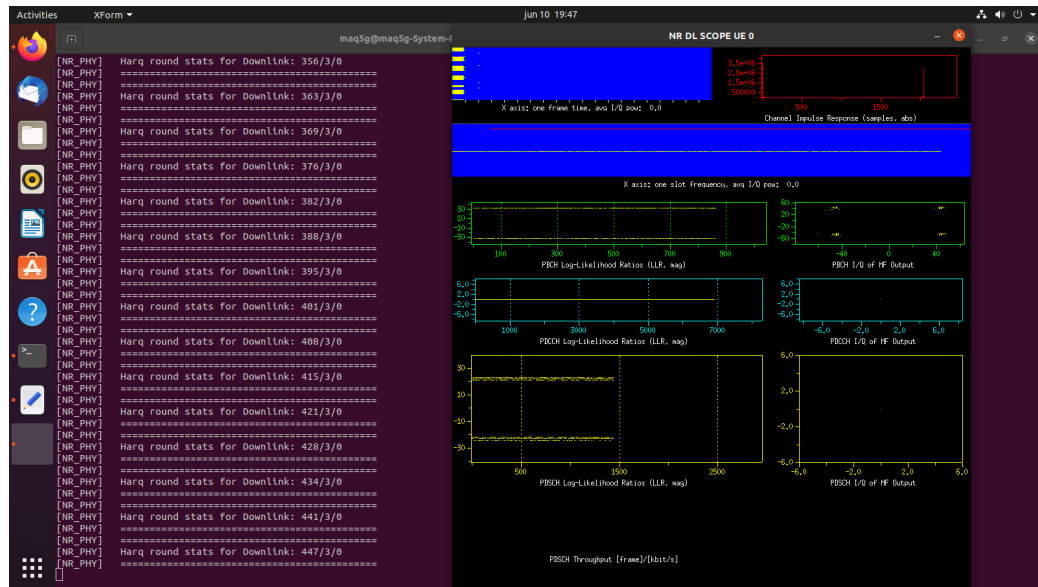


Figura A.6: Gráficos UE estatus

Se corre el UE con el comando:

```
sudo RFSIMULATOR=192.168.10.3 ./nr-uesoftmodem -r 106 --numerology  
1 --band 78 -C 3619200000 --nokrnmod --rfsim --sa --uicc0.imsi  
001010000000001 --uicc0.nssai_sd 1 -d
```

La opción `-d` al final del comando, despliega un gráfico indicativo del estado de la comunicación y como varían los parámetros en tiempo real, figuras: A.6 y A.7 . En el se puede apreciar en ambos casos, una trama en el dominio del tiempo en donde el color representa la energía promedio, además de un gráfico con la respuesta al impulso. En el gráfico del UE se despliegan los canales Physical Broadcast Channel (PBCH), Physical Downlink Common Control Channel (PDCCH) y Physical Downlink Shared Channel (PDSCH), ambos con los radios en el dominio temporal y las constelaciones de las señales luego de la ecualización en fase y cuadratura para cada uno de los canales físicos.

se accede al shell de comandos del UE1 tipeando en la ventana
del terminal:

```
ip netns exec ue1 bash
```

Se debe levantar el tunel desde el UE al core agregando la ruta:

```
sudo route add default gw 12.1.1.1
```

Ping para confirmar conectividad, desde la terminal donde fue
creado cada UE al core:

Apéndice A. Manuales

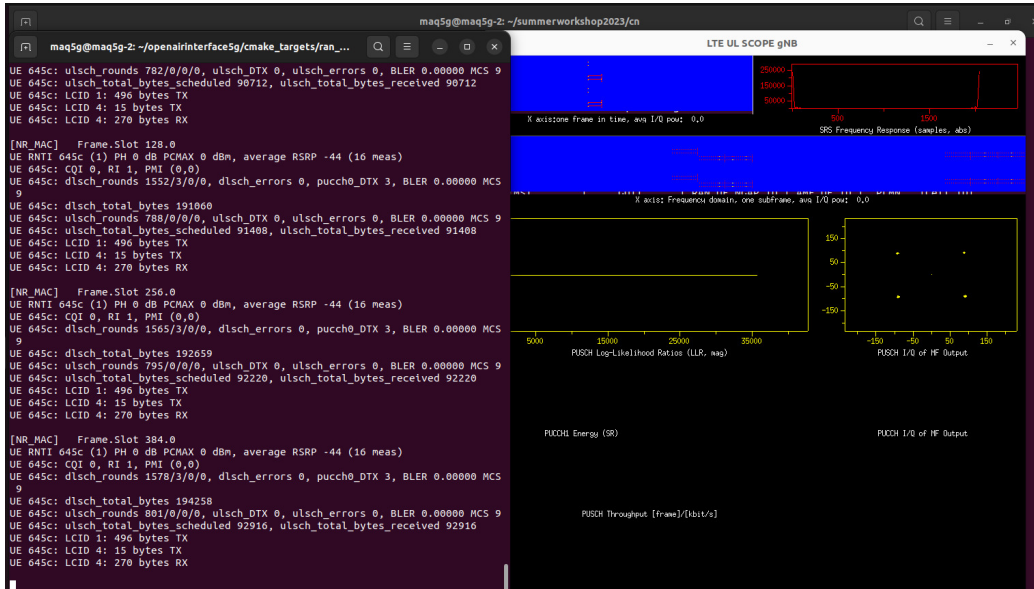


Figura A.7: Gráficos gNB estatus

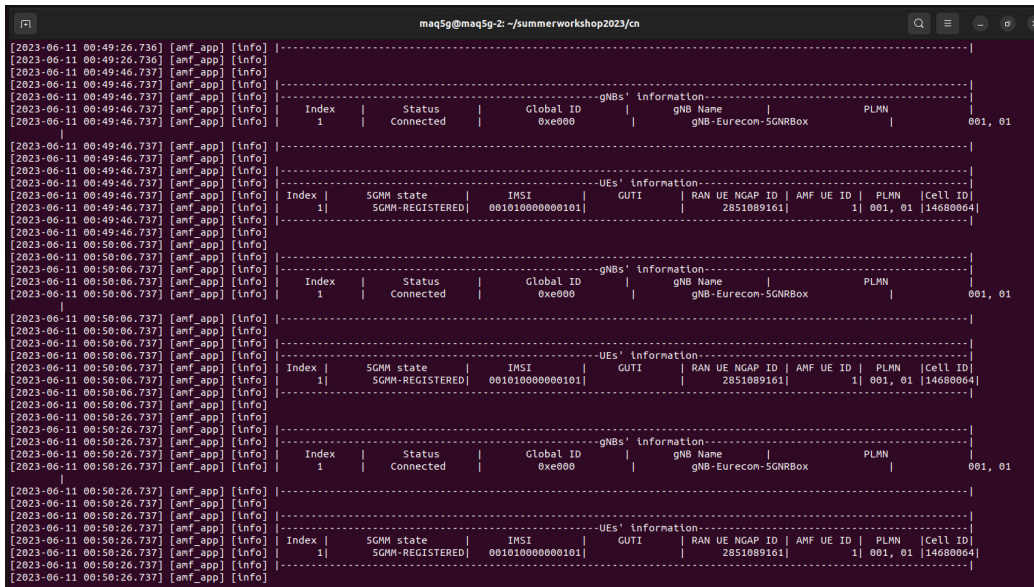


Figura A.8: Logs Core AMF

A.2. Instalación de la plataforma

```
ping 192.168.70.135 -I oaitun_ue1

# IPERF para medir ancho de banda, cliente en el UE y servidor en
el core:

iperf3 -B 12.1.1.151 -c 192.168.70.135 -u -R -b50M
```

Por último en la figura: A.8, se muestran los logs del Core AMF, observamos el gNB conectado y la lista de UE registrados con su correspondiente IMSI.

Contenido del script utilizado

Script compartido en el Git del OAI.

```
#!/bin/bash

ue_id=-1

create_namespace() {
    ue_id=$1
    local name="ue$ue_id"
    echo "creating namespace for UE ID ${ue_id} name ${name}"
    ip netns add $name
    ip link add v-eth$ue_id type veth peer name v-ue$ue_id
    ip link set v-ue$ue_id netns $name
    BASE_IP=$((200+ue_id))
    ip addr add 10.$BASE_IP.1.100/24 dev v-eth$ue_id
    ip link set v-eth$ue_id up
    iptables -t nat -A POSTROUTING -s 10.$BASE_IP.1.0/255.255.255.0 -o lo
-j MASQUERADE
    iptables -A FORWARD -i lo -o v-eth$ue_id -j ACCEPT
    iptables -A FORWARD -o lo -i v-eth$ue_id -j ACCEPT
    ip netns exec $name ip link set dev lo up
    ip netns exec $name ip addr add 10.$BASE_IP.1.$ue_id/24 dev v-ue$ue_id
    ip netns exec $name ip link set v-ue$ue_id up
}

delete_namespace() {
    local ue_id=$1
    local name="ue$ue_id"
    echo "deleting namespace for UE ID ${ue_id} name ${name}"
    ip link delete v-eth$ue_id
    ip netns delete $name
}
```

Apéndice A. Manuales

```
list_namespaces() {
    ip netns list
}

open_namespace() {
    if [[ $ue_id -lt 1 ]]; then echo "error: no last UE processed"; exit 1; fi
    local name="ue$ue_id"
    echo "opening shell in namespace ${name}"
    echo "type 'ip netns exec $name bash' in additional terminals"
    ip netns exec $name bash
}

usage () {
    echo "$1 -c <num>: create namespace \"ue<num>\""
    echo "$1 -d <num>: delete namespace \"ue<num>\""
    echo "$1 -e      : execute shell in last processed namespace"
    echo "$1 -l      : list namespaces"
    echo "$1 -o <num>: open shell in namespace \"ue<num>\""
}

prog_name=$(basename $0)

if [[ $(id -u) -ne 0 ]] ; then echo "Please run as root"; exit 1; fi
if [[ $# -eq 0 ]]; then echo "error: no parameters given"; usage $prog_name;
exit 1; fi

while getopts c:d:ehlo: cmd
do
    case "${cmd}" in
        c) create_namespace ${OPTARG};;
        d) delete_namespace ${OPTARG};;
        e) open_namespace; exit;;
        h) usage ${prog_name}; exit;;
        l) list_namespaces;;
        o) ue_id=${OPTARG}; open_namespace;;
        /?) echo "Invalid option"; usage ${prog_name}; exit;;
    esac
done
```

A.3. Procedimientos

A.3.1. Instalar y Correr un Sistema Completo en Dockers

Se muestra una instalación ágil y preconfigurada que permite utilizar el sistema con una configuración fija. Se utiliza para ello un sistema compartido para el workshop de mayo de 2023.

Instalar en Ubuntu 20.04 o 22.04.

La computadora debe soportar avx2, verificar con:

```
lscpu | grep avx2
```

Condiciones previas:

- docker engine, Mínimo 19.03.6
- docker-compose, Mínimo 1.27.4

Nota: si la versión de docker es mayor a 21, el docker-compose está incluido.

Crear el entorno:

```
mkdir -p /tmp/oaisummerworkshop2023
cd /tmp/oaisummerworkshop2023
git clone https://gitlab.eurecom.fr/oaiworkshop/summerworkshop2023.git
cd /tmp/oaisummerworkshop2023/summerworkshop2023/cn
```

Extraer las imágenes de los contenedores para el CN, el gNB y el UE.

```
##Recommended to login to your docker hub account
docker login
## Pull images
docker pull oaisoftwarealliance/oai-amf:v1.5.1
docker pull oaisoftwarealliance/oai-smf:v1.5.1
docker pull oaisoftwarealliance/oai-nrf:v1.5.1
docker pull oaisoftwarealliance/oai-udr:v1.5.1
docker pull oaisoftwarealliance/oai-udm:v1.5.1
docker pull oaisoftwarealliance/oai-ausf:v1.5.1
docker pull oaisoftwarealliance/oai-spgwu-tiny:v1.5.1
## Descargar imágenes de gNB y UE
docker pull oaisoftwarealliance/oai-nr-ue:2023.w19
docker pull oaisoftwarealliance/oai-gnb:2023.w19
```

Si se quiere correr el CN y la RAN en máquinas diferentes se debe habilitar la retransmisión de paquetes IP.

```
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo iptables -P FORWARD ACCEPT
# También se precisa agregar una ruta estática en la máquina del gNB para
# que llegue al CN.
sudo ip route add route -net SUBRED-DEL-CN-/MASCARA gw IP-PROXIMO-SALTO
```

Crear la interfaz de red con los dockers.

Apéndice A. Manuales

```
docker network create --driver=bridge --subnet=192.168.70.128/26 -o
"com.docker.network.bridge.name"="oaiworkshop" oaiworkshop
```

Iniciar el CN.

```
cd summerworkshop2023/cn
docker compose -f docker-compose.yml up -d
```

Esperar a que todos los contenedores queden *healthy*. Se puede verificar con:

```
docker-compose -f docker-compose.yml ps -a
```

Dejar corriendo el log del AMF con:

```
docker logs oai-amf -f
```

En otro terminal dejar corriendo el log del SMF con:

```
docker logs oai-smf -f
```

En otro terminal correr un gNB. Verificar en el log del gNB que corre y queda operativo, y en el del AMF que queda registrado.

```
docker compose -f docker-compose-ran.yml up -d oai-gnb
```

En otro terminal correr un UE. Verificar que queda corriendo en su log. Se puede ver la conexión en el log del gNB, con los reportes de estado de la conexión, y en el log del AMF con el registro del UE.

```
docker compose -f docker-compose-ran.yml up -d oai-nr-ue
```

Una vez corriendo el UE, se puede pasar tráfico haciendo ping a la interfaz del túnel GTPU con la IP 12.1.1.1, o al generador de tráfico interno del CN, 192.168.70.135, o, si se tiene acceso internet, a una ip externa, como por ejemplo la 8.8.8.8 del DNS de Google. También se pueden hacer pruebas de IPERF al generador de tráfico o a otro servidor colocado en redes externas.

```
#Posicionarse en la consola del UE
docker exec -it oai-nr-ue bash
## ping por internet
ping -I oaitun_ue1 8.8.8.8 -c4
## ping al upf/spgwn
ping -I oaitun_ue1 12.1.1.1 -c4
## ping tal generador interno oai-traffic-generator
ping -I oaitun_ue1 192.168.70.135 -c4
## Prueba de Iperf con el server del generador interno
iperf3 -B <UE_IP_ADDRESS> -c 192.168.70.135 -u -R -b20M
```

Es posible correr un segundo UE ejecutando:

```
docker compose -f docker-compose-ran.yml up -d oai-nr-ue2
```

Y se pueden repetir las mismas pruebas o hacer pruebas de un UE al otro. Para cerrar, desde la carpeta de donde se ejecutaron se debe correr:

```
docker stop oai-nr-ue oai-nr-ue2
docker stop oai-nr-gnb
```

Y si se quiere parar el CN:

```
docker compose -f docker-compose.yml down -t2
```

A.3.2. Archivo de Configuración Comentado

Se presenta a continuación el archivo de configuración utilizado para el gNB de 40 MHz, 106 PRB. Se mantienen los comentarios originales en inglés del template y se agregan nuestros comentarios en español. La presente fue una configuración muy utilizada en las pruebas de este trabajo. Corresponde al archivo de nombre gnb.sa.band78.106prb.rfsim.alfabeta.conf.

```
Active_gNBs = ( "gNB-FING-MAQ5G");
# Asn1\_verbosity, choice in: none, info, annoying
Asn1_verbosity = "none";
sa=1; // Sistema Stand Alone, si no está este comando, no se genera el
RACH y se intenta levantar interfaz X2 con eNB ancla.

device: { name = "rfsimulator" }

gNBs =
(
{
////////// Identification parameters: //ID y nombre para el AMF
gNB_ID = 0xe01;
gNB_name = "gNB-FING-MAQ5G";

// Tracking area code, 0x0000 and 0xffff are reserved values
//debe coincidir con el definido en el AMF
tracking_area_code = 1;

// el PLMN debe coincidir con las definiciones del CN,
plmn_list =({mcc =001; mnc =01; mnc_length =2; sncsaiList =({ sst =1})});

nr_cellid = 12345678L;

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

ssb_SubcarrierOffset = 0;
// número de puertos lógicos de antena de tx -layers
pdsch_AntennaPorts_N1 = 1;
//número de puertos lógicos de antena de rx -layers
pusch_AntennaPorts = 1;
//tiempo en slots entre pdsch y pusch.
min_rxtxtime = 6; // usar 6 para UL OAI y 2 para teléfonos
enable_sdap = 0;
#sib1_tda = 0;

pdccch_ConfigSIB1 = (
{
controlResourceSetZero = 12;
searchSpaceZero = 0;
}
);

servingCellConfigCommon = (
```

Apéndice A. Manuales

```
{
#spCellConfigCommon

    physCellId = 0;

# downlinkConfigCommon
#frequencyInfoDL
    # this is 3600 MHz + 43 PRBs@30kHz SCS (same as initial BWP)
    absoluteFrequencySSB = 641280; //frecuencia central
//del SSB, debe ser parte del raster de sincronismo.
    dl_frequencyBand = 78; //Banda de operación
    # this is 3600 MHz
    dl_absoluteFrequencyPointA = 640008; //ARFCN de
// la primera subportadora del ancho de banda total,
//independientemente de la BWP en uso
    #scs-SpecificCarrierList
        dl_offstToCarrier = 0; //separación en Hz
//entre Tx y Rx, el 0 indica TDD.
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    dl_subcarrierSpacing = 1;
    dl_carrierBandwidth = 106; // Ancho de banda en PRB
#initialDownlinkBWP
#genericParameters
    # this is RBstart=27,L=48 (275*(L-1))+RBstart
    initialDLBWPlocationAndBandwidth = 28875; # 6366 12925 12956 28875
//12952 //28875 es el resultado de la fórmula anterior usando RBstart =0,
//es decir comenzando en la primera subportadora, y L=106, es decir
//asignando todo el ancho de banda.
# subcarrierSpacing /
// se usan los mismos parámetros para la BWP inicial y para las
//configuraciones de UL
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    initialDLBWPsubcarrierSpacing = 1;
#pdccch-ConfigCommon
    initialDLBWPcontrolResourceSetZero = 12;
    initialDLBWPsearchSpaceZero = 0;

# uplinkConfigCommon
# frequencyInfoUL
    ul_frequencyBand = 78;
    #scs-SpecificCarrierList
        ul_offstToCarrier = 0;
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    ul_subcarrierSpacing = 1;
    ul_carrierBandwidth = 106;
    pMax = 20;
#initialUplinkBWP
#genericParameters
initialULBWPlocationAndBandwidth = 28875;
```


A.3. Procedimientos

```

# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    initialULBWPsubcarrierSpacing = 1;
#rach-ConfigCommon
#rach-ConfigGeneric
    prach_ConfigurationIndex = 98;
#prach_msg1_FDM
#0 = one, 1=two, 2=four, 3=eight
    prach_msg1_FDM = 0;
    prach_msg1_FrequencyStart = 0;
    zeroCorrelationZoneConfig = 13;
    preambleReceivedTargetPower = -96;
#preambleTransMax (0..10) = (3,4,5,6,7,8,10,20,50,100,200)
    preambleTransMax = 6;
#powerRampingStep
# 0=dB0,1=dB2,2=dB4,3=dB6
    powerRampingStep = 1;
#ra_ReponseWindow
#1,2,4,8,10,20,40,80
    ra_ResponseWindow = 4;
#ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR
#1=oneeighth,2=onefourth,3=half,4=one,5=two,6=four,7=eight,8=sixteen
    ssb_perRACH_OccasionAndCB_PreamblesPerSSB_PR = 4;
#oneHalf (0..15) 4,8,12,16,...60,64
    ssb_perRACH_OccasionAndCB_PreamblesPerSSB = 14;
#ra_ContentionResolutionTimer
#(0..7) 8,16,24,32,40,48,56,64
    ra_ContentionResolutionTimer = 7;
    rsrp_ThresholdSSB = 19;
#prach-RootSequenceIndex_PR
#1 = 839, 2 = 139
    prach_RootSequenceIndex_PR = 2;
    prach_RootSequenceIndex = 1;
    # SCS for msg1, can only be 15 for 30 kHz < 6 GHz,
    #takes precedence over the one derived from prach-ConfigIndex
    #
    msg1_SubcarrierSpacing = 1,
# restrictedSetConfig
# 0=unrestricted, 1=restricted type A, 2=restricted type B
    restrictedSetConfig = 0,

    msg3_DeltaPreamble = 1;
    p0_NominalWithGrant = -90;

# pucch-ConfigCommon setup :
# pucchGroupHopping
# 0 = neither, 1= group hopping, 2=sequence hopping
    pucchGroupHopping = 0;
    hoppingId = 40;
    p0_nominal = -90;
# ssb_PositionsInBursts_BitmapPR

```

Apéndice A. Manuales

```
# 1=short, 2=medium, 3=long
    ssb_PositionsInBurst_PR = 2;
    ssb_PositionsInBurst_Bitmap = 1;

# ssb_periodicityServingCell
# 0 = ms5, 1=ms10, 2=ms20, 3=ms40, 4=ms80, 5=ms160, 6=spare2, 7=spare1
//frecuencia con que se repite el SSB
    ssb_periodicityServingCell = 2;

# dmrs_TypeA_position
# 0 = pos2, 1 = pos3
    dmrs_TypeA_Position = 0;

# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    subcarrierSpacing = 1;

#tdd-UL-DL-ConfigurationCommon
// Conjunto de definiciones del armado del patrón de slots en TDD
# subcarrierSpacing
# 0=kHz15, 1=kHz30, 2=kHz60, 3=kHz120
    referenceSubcarrierSpacing = 1;
# pattern1
# dl_UL_TransmissionPeriodicity
# 0=ms0p5, 1=ms0p625, 2=ms1, 3=ms1p25, 4=ms2, 5=ms2p5, 6=ms5, 7=ms10
    dl_UL_TransmissionPeriodicity = 6;
    nrofDownlinkSlots = 7;
    nrofDownlinkSymbols = 6;
    nrofUplinkSlots = 2;
    nrofUplinkSymbols = 4;

    ssPBCH_BlockPower = -25;
}

);

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

////////// AMF parameters:
amf_ip_address = ( { ipv4 = "192.168.70.132";
                    ipv6 = "192:168:30::17";
                    active = "yes";
```

A.3. Procedimientos

```
        preference = "ipv4";
    }
);

NETWORK_INTERFACES :
{
//interfaz virtual entre la red de dockers y la máquina host
    GNB_INTERFACE_NAME_FOR_NG_AMF        = "demo-oai";
//ip de la máquina host en la red del Core donde se ubica el AMF
    GNB_IPV4_ADDRESS_FOR_NG_AMF         = "192.168.70.129/26";
    GNB_INTERFACE_NAME_FOR_NGU          = "demo-oai";
    GNB_IPV4_ADDRESS_FOR_NGU            = "192.168.70.129/26";
    GNB_PORT_FOR_S1U                     = 2152; # Spec 2152
};

}
);

MACRLCs = (
    {
        num_cc                = 1; //cantidad de portadoras
        tr_s_preference        = "local_L1"; //interfaz de capa física
        tr_n_preference        = "local_RRC"; //interfaz con RRC
        pusch_TargetSNRx10     = 150;
//valor objetivo de la SNR en el canal compartido=15 dB
        pucch_TargetSNRx10     = 200;
//valor objetivo de la SNR en el canal de control=20 dB
        ulsch_max_frame_inactivity = 20;
//cada 20 tramas se asignan 5 PRB al UE, para mantener la conexión activa
        dl_harq_round_max      = 4; //Máximo de retransmisiones en DL
        ul_harq_round_max      = 4; //Máximo de retransmisiones en UL
        dl_max_mcs              = 28; // valor máximo de mcs en DL
        ul_max_mcs              = 28; // valor máximo de mcs en UL
        #Se agregan exponentes alfa y beta en UL y DL como operadores
        #para variar el scheduler
        # PF 1 y 1
        # MaxTBS 1 y 0
        # Round Robin 0 y 0

        ul_alfa = 1; # por defecto 1, exponente del tbs en numerador
        ul_beta = 1; # por defecto 1, exponente del throughput en denominador
        dl_alfa = 1; # por defecto 1, exponente del tbs en numerador
        dl_beta = 1; # por defecto 1, exponente del throughput en denominador
    }
);

L1s = (
    {
num_cc = 1; //cantidad de portadoras en capa física
tr_n_preference = "local_mac";
```

Apéndice A. Manuales

```
prach_dtx_threshold = 120;
    pucch0_dtx_threshold = 150;
    ofdm_offset_divisor = 8; #set this to UINT_MAX for offset 0
}
);

RUs = (
    {
        local_rf      = "yes" //Se espera un SDR
        nb_tx         = 1 //cantidad de antenas fisicas de tx
        nb_rx         = 1 //cantidad de antenas fisicas de rx
        att_tx        = 0
//Valor fijo de atenuación agregada por sw al transmisor
        att_rx        = 0;
//Valor fijo de atenuación agregada por sw al receptor
        bands         = [78];
        max_pdschReferenceSignalPower = -27;
        max_rxgain    = 114; //Límite de ganancia
//máxima en Rx.Un límite bajo deja afuera UE con mala señal,
//un límite alto empeora la SNR.
        eNB_instances = [0]; // se habilita en NSA
        #beamforming 1x4 matrix:
        bf_weights = [0x00007fff, 0x0000, 0x0000, 0x0000];
        clock_src = "internal"; // origen del sincronismo para el reloj
//del SDR, interno (funciona libre), externo (cable de entrada) o gps.
    }
);

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

rfsimulator :
{
    serveraddr = "server";
    serverport = "4043";
    options = (); #("saviq"); or/and "chanmod"
    modelname = "AWGN";
    IQfile = "/tmp/rfsimulator.iqs";
};

security = {
    # preferred ciphering algorithms
    # the first one of the list that an UE supports in chosen
    # valid values: nea0, nea1, nea2, nea3
```

```

ciphering_algorithms = ( "nea0" );

# preferred integrity algorithms
# the first one of the list that an UE supports is chosen
# valid values: nia0, nia1, nia2, nia3
integrity_algorithms = ( "nia2", "nia0" );

# setting 'drb_ciphering' to "no" disables ciphering for DRBs, no matter
# what 'ciphering_algorithms' configures; same thing for 'drb_integrity'
drb_ciphering = "yes";
drb_integrity = "no";
};

//nivel de mensajes en log del gNB, se puede elegir entre
//error, warn, analysis, info, debug y trace para cada
//conjunto de mensajes
log\_config = {
    global_log_level           = "info";
    hw_log_level               = "info";
    nr_phy_log_level           = "info";
    nr_mac_log_level           = "info";
    rlc_log_level              = "info";
    pdcp_log_level             = "info";
    rrc_log_level              = "info";
    gtp_log_level              = "info";
    flap_log_level             = "info";
    ngap_log_level             = "info";
};

#@include "channelmod_rfsimu.conf"
//se descomenta para agregar AWGN en la señal de UL cuando se usa
con RFSim.

```

A.3.3. Cambiar Banda, Frecuencia o Ancho de Banda en un gNB

Para cambiar la banda, la frecuencia y/o el ancho de banda en un gNB se debe modificar el archivo de configuración utilizado. Las explicaciones a continuación se hacen sobre el archivo de configuración presentado en la sección A.3.2.

Las frecuencias están expresadas por su *Absolute Radio Frequency Channel Number* (ARFCN) en el archivo de configuración. Son números de canal separados cada 5 kHz para frecuencias entre 0 y 3000 MHz, siendo los $ARFCN \in [0, 599999]$. Para frecuencias en el rango (3000 MHz, 24 250 MHz) los canales están separados 15 kHz y usan los valores [600 000, 2 016 666]. En frecuencias mayores el espaciamiento es de 60 kHz, pero no se utilizan en la maqueta. En cambio la diferencia de frecuencia entre transmisión y recepción en configuraciones FDD se expresa en Hertz. Los anchos de banda se expresan en PRB y dependen de la numerología inicial.

El bloque de sincronismo debe tener su frecuencia central coincidente con un

Apéndice A. Manuales

Global Synchronization Channel Number (GSCN). Estos canales están separados 1.44 MHz, lo que corresponde a 96 canales ARFCN en la banda 78. Un UE solo busca el bloque de sincronismo en estos canales por razones de velocidad de la búsqueda.

Los parámetros a configurar en el gNB se separan en DL y UL. En DL son:

- `absoluteFrequencySSB`: ARFCN central del bloque de sincronismo SSB. No es cualquier valor. Debe pertenecer a la banda de frecuencia declarada, y además ser parte del raster de sincronismo. Los 20 PRB de ancho del SSB deben quedar dentro de la banda, dentro del ancho de banda configurado y dentro de la BWP inicial.
- `dl_frequencyBand`: Banda de frecuencia a utilizar.
- `dl_absoluteFrequencyPointA`: ARFCN de la primera subportadora del canal. Es la frecuencia central de esa subportadora.
- `dl_offsetToCarrier`: Si la banda elegida es de uso TDD, debe ser 0. Si es de uso FDD debe decir la separación en Hz entre transmisión y recepción.
- `dl_subcarrierSpacing`: Numerología utilizada en la condición inicial. Se pueden declarar otras BWP con numerologías diferentes.
- `dl_carrierBandwidth`: Ancho de banda en PRB. El SSB y las BWP deben quedar todos dentro de la banda definida por el Point A y este ancho.
- `initialDLBWPlocationAndBandwidth`: punto de inicio y tamaño total de la BWP inicial en DL dado como Resource Indication Value (RIV). Es un número que se calcula con ecuaciones que dependen del tamaño de la BWP L dado en PRB. Si $L - 1 \leq \lfloor \frac{N_{BWP}^{size}}{2} \rfloor$ entonces

$$RIV = (N_{BWP}^{size} \times (L - 1)) + RB_{start}$$

y si es mayor, entonces

$$RIV = N_{BWP}^{size} \times (N_{BWP}^{size} - L + 1) + (N_{BWP}^{size} - 1 - RB_{start})$$

donde $N_{BWP}^{size} = 275$ que es el número máximo posible de PRB, L es el largo deseado de la BWP y RB_{start} es el resource block inicial.

Para el UL los parámetros son los mismos. En este trabajo siempre se eligió el RIV para que la BWP inicial coincidiera con el ancho de banda total, y fue la única utilizada.

Se recomienda utilizar una calculadora web para las conversiones de frecuencia a ARFCN y cálculos de RIV como 5G-Tools [7] o sqimway [37].

Una vez configurado el gNB, un teléfono debe poder encontrarlo haciendo la búsqueda del bloque de sincronismo. Pero el UE de OAI no hace este proceso. Requiere que se le indique en el comando, la frecuencia del SSB en Hz, la separación entre UL y DL si es FDD, el ancho de banda, la banda y el offset entre la primera subportadora y el SSB.

Como ejemplo, para correr un UE que sincronice con este gNB el comando es:

A.3. Procedimientos

```
sudo ./nr-uesoftmodem -r 106 --numerology 1 --band 78 -C 3619200000
--ssb 516 -E --sa --rfsim --uicc0.imsi 001010000000101
--rfsimulator.serveraddr 192.168.40.3 -d
```

Donde:

- r 106 indica el ancho de banda
- band 78 es la banda
- C 3619200000 es la frecuencia en Hertz correspondiente. Es la frecuencia del absoluteFrequencySSB en el gNB, pasada a Hertz
- ssb 516 indica el offset entre la frecuencia de punto A (primera subportadora) y el inicio del SSB. En este caso podría estar omitido, porque el valor usado de 516 es el valor por defecto del sistema.

Se ejemplifica el cálculo del offset de SSB para el UE con este caso. La configuración del gNB está hecha para 106 PRB, 40 MHz, FR1 banda 78 e incluye:

- absoluteFrequencySSB = 641280 (GSCN = 7929, 3 619 200 000 Hz) Frecuencia central del SSB
- dlAbsoluteFrequencyPointA = 640008 (3 600 120 000 Hz)

La diferencia en frecuencia entre el punto medio y el inicio del SSB es

$$\frac{SSB}{2} = \frac{20 \times PRB \times 12 \times 30kHz}{2} = 3\,600\,000Hz$$

Luego el offset entre la primera subportadora, y la primera subportadora del SSB es:

$$Offset = \frac{absoluteFrequencySSB - \frac{SSB}{2} - dl_absoluteFrequencyPointA}{SCS}$$

Que puesto en los valores del ejemplo es:

$$\frac{3\,619\,200\,000 - 3\,600\,000 - 3\,600\,120\,000}{30\,000} = 516$$

Como ayuda se puede encontrar el valor de la frecuencia a poner en el UE en el log del gNB, al inicio:

```
fp->dl_CarrierFreq=3619200000
[PHY] fp->ul_CarrierFreq=3619200000
[PHY] Setting RF config for N_RB 106, NB_RX 1, NB_TX 1
[PHY] Channel 0: setting tx_gain offset 0, rx_gain offset 114,
tx_freq 3619200000 Hz, rx_freq 3619200000 Hz, tune_offset 0 Hz,
sample_rate 46080000 Hz
```

Apéndice A. Manuales

También es posible leer el offset entre la frecuencia del punto A y el SSB ejecutando el gNB con el siguiente comando:

```
sudo -E RFSIMULATOR=server ./nr-softmodem --rfsim --sa -0
~/Desktop/conf/gnb.sa.band78.106prb.rfsim.alfabeta.conf
--log_config.phy_log_level debug | grep "SSB first"
```

El último parámetro habilita a enviar todos los mensajes de debug preestablecidos de la capa física a la consola estándar y luego se filtran las líneas que incluyen el valor buscado. En la consola del gNB se verá el valor del SSB requerido en un UE.

A.3.4. Procedimiento de Pruebas de SDR

Durante el presente trabajo se encontró una daughterboard UBX-160 que no transmite y debió ser remplazada. En este procedimiento se describen la secuencia de pruebas que permite verificar un SDR utilizando solamente las herramientas de software incluidas en su driver UHD.

Condiciones iniciales:

- driver UHD instalado
- SDR ETTUS ensamblado
- Antenas conectadas correctamente
- conexión ethernet o USB 3.0 con la PC

Se prueba la conexión y se revisa el inventario instalado con el comando `uhd_usrp_probe`. En el resultado se tiene el reporte del equipo conectado, su inventario, configuración básica y las placas agregadas,

```
maq5g:/usr/local/lib/uhd/examples$ uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 9.4.0; Boost_107100; UHD_4.3.0.HEAD-0
[INFO] [X300] X300 initialization sequence...
[INFO] [X300] Maximum frame size: 1472 bytes.
[INFO] [X300] Radio 1x clock: 200 MHz
[WARNING] [RFNOC::GRAPH] One or more blocks timed out during flush!
```

```
-----
/
|       Device: X-Series Device
|       -----
|       /
|       |       Mboard: X310
|       |       revision: 11
|       |       revision_compat: 7
|       |       product: 30818
|       |       mac-addr0: 00:80:2f:18:37:2b
|       |       mac-addr1: 00:80:2f:18:37:2c
|       |       gateway: 192.168.10.1
|       |       ip-addr0: 192.168.10.2
```


A.3. Procedimientos

```
| | subnet0: 255.255.255.0
| | ip-addr1: 192.168.20.2
| | subnet1: 255.255.255.0
| | ip-addr2: 192.168.30.2
| | subnet2: 255.255.255.0
| | ip-addr3: 192.168.40.2
| | subnet3: 255.255.255.0
| | serial: 3138E7B
| | FW Version: 6.0
| | FPGA Version: 39.0
| | FPGA git hash: c781f75
| | RFNoC capable: Yes
| |
| | Time sources: internal, external, gpsdo
| | Clock sources: internal, external, gpsdo
| | Sensors: ref_locked
| |
| |-----
| | /
| | RFNoC blocks on this device:
| |
| | * 0/DDC#0
| | * 0/DDC#1
| | * 0/DUC#0
| | * 0/DUC#1
| | * 0/Radio#0
| | * 0/Radio#1
| | * 0/Replay#0
| |
| |-----
| | /
| | Static connections on this device:
| |
| | * 0/SEP#0:0==>0/DUC#0:0
| | * 0/DUC#0:0==>0/Radio#0:0
| | * 0/Radio#0:0==>0/DDC#0:0
| | * 0/DDC#0:0==>0/SEP#0:0
| | * 0/Radio#0:1==>0/DDC#0:1
| | * 0/DDC#0:1==>0/SEP#1:0
| | * 0/SEP#2:0==>0/DUC#1:0
| | * 0/DUC#1:0==>0/Radio#1:0
| | * 0/Radio#1:0==>0/DDC#1:0
| | * 0/DDC#1:0==>0/SEP#2:0
| | * 0/Radio#1:1==>0/DDC#1:1
| | * 0/DDC#1:1==>0/SEP#3:0
| | * 0/SEP#4:0==>0/Replay#0:0
| | * 0/Replay#0:0==>0/SEP#4:0
| | * 0/SEP#5:0==>0/Replay#0:1
| | * 0/Replay#0:1==>0/SEP#5:0
| |
| |-----
| | /
| | TX Dboard: 0/Radio#0
| | ID: UBX-160 v2 (0x007d)
```

Apéndice A. Manuales

```
| | Serial: 3120D16
| | Revision: 4
| |
| | -----
| | /
| | | TX Frontend: 0
| | | Name: UBX TX
| | | Antennas: TX/RX, CAL
| | | Sensors: lo_locked
| | | Freq range: 10.000 to 6000.000 MHz
| | | Gain range PGA0: 0.0 to 31.5 step 0.5 dB
| | | Bandwidth range: 160000000.0 to 160000000.0 step 0.0 Hz
| | | Connection Type: QI
| | | Uses LO offset: No
| |
| | -----
| | /
| | | RX Dboard: 0/Radio#0
| | | ID: UBX-160 v2 (0x007e)
| | | Serial: 3120D16
| | | Revision: 4
| |
| | -----
| | /
| | | RX Frontend: 0
| | | Name: UBX RX
| | | Antennas: TX/RX, RX2, CAL
| | | Sensors: lo_locked
| | | Freq range: 10.000 to 6000.000 MHz
| | | Gain range PGA0: 0.0 to 31.5 step 0.5 dB
| | | Bandwidth range: 160000000.0 to 160000000.0 step 0.0 Hz
| | | Connection Type: IQ
| | | Uses LO offset: No
| |
| | -----
| | /
| | | TX Dboard: 0/Radio#1
| |
| | -----
| | /
| | | TX Frontend: 0
| | | Name: Unknown (0xffff) - 0
| | | Antennas:
| | | Sensors:
| | | Freq range: 0.000 to 0.000 MHz
| | | Gain Elements: None
| | | Bandwidth range: 0.0 to 0.0 step 0.0 Hz
| | | Connection Type: IQ
| | | Uses LO offset: No
| |
| | -----
| | /
| | | RX Dboard: 0/Radio#1
| |
| | -----
| | /
| | | RX Frontend: 0
| | | Name: Unknown (0xffff) - 0
```

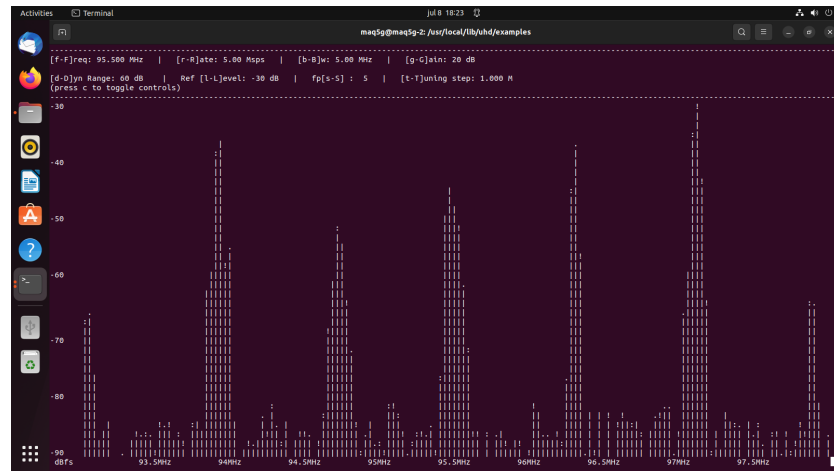


Figura A.9: Espectro de radioemisora a 95.5 MHz y entorno usando la aplicación dft del driver UHD.

```
| | | Antennas:
| | | Sensors:
| | | Freq range: 0.000 to 0.000 MHz
| | | Gain Elements: None
| | | Bandwidth range: 0.0 to 0.0 step 0.0 Hz
| | | Connection Type: IQ
| | | Uses LO offset: No
```

Una vez que se confirma que todas las placas están funcionando se prueba la recepción, viendo señales conocidas, por ejemplo el espectro en la banda de FM. Se utiliza para esto el programa incluido en la carpeta de ejemplos del driver uhd/examples/rx_ascii_art_dft, con el comando:

```
./rx_ascii_art_dft --freq 95.5e6 --rate 5e6 --gain 20 --bw 5e6 --ref-lvl
-30
```

Que muestra como salida a la figura A.9.

Una vez verificada la recepción, corresponde verificar la transmisión. Para ello se necesita un segundo SDR con antenas, que se haya verificado su recepción. Se utiliza un generador de senoide para transmitir y se observa con el otro SDR con la aplicación rx_ascii_art_dft ya utilizada. En el SDR bajo prueba se ejecuta:

```
./tx_waveforms --freq 1e9 --rate 5e6 --gain 0
```

y en el segundo SDR, con el comando

```
./rx_ascii_art_dft --freq 1e9 --rate 5e6 --gain 20 --bw 5e6
--ref-lvl -30
```

Se observa la figura A.10 que confirma la transmisión.

Apéndice A. Manuales

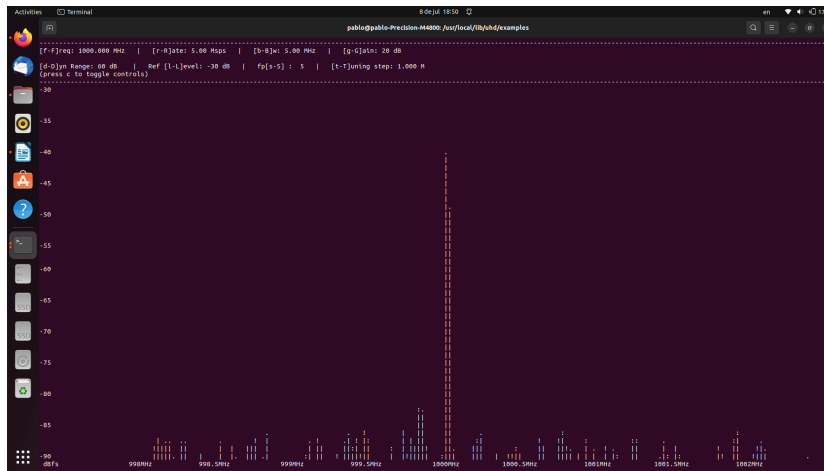


Figura A.10: Señal de 1 GHz transmitida por el SDR bajo prueba y verificada por un segundo SDR, con la herramienta del driver rx_ascii_art_dft.

A.4. Herramientas Adicionales

Las medidas de throughput se hicieron usando iperf3.

Las capturas de tráfico para el análisis de protocolos se hicieron utilizando Wireshark 3.6.2.

Como analizador de espectro se utilizó un SDR B200 con el software gqrx.

Apéndice B

Instalación de Plataforma srsRAN

B.1. Instalación de radios virtuales

Se parte de la base, de que se tiene instalado Ubuntu 20.04 server. Se ejecuta la instalación siguiendo estos pasos, extraídos de las indicaciones de srsRAN en [38] y [40].

1. Instalar librerías ZMQ: `sudo apt-get install libzmq3-dev`
2. Instalar srsRAN software:
`sudo add-apt-repository ppa:softwareradiosystems/srsran`
3. En `/root/.config/srsran`, copiar los archivos de configuración que pertenecen al ZMQ, `ue.conf`, `rr.conf` y `enb.conf`. En particular el archivo `enb.conf`, se modifica de la forma mostrada en la figura B.1 para que funcione como NR.

```
NR configuración
carrier(s):

[rat.eutra]
dl_earfcn = 2850
nof_carriers = 0

NR band(s) and carrier(s):

[rat.nr]
bands = 3,78
nof_carriers = 1

release-15:

[rrc]
release = 15
```

Figura B.1: NR configuración.

B.2. Instalación del core Open5gs

```
1) instalar mongo db:
$ sudo apt update
$ sudo apt install wget gnupg
$ wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc — sudo apt-
key add -
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-
org/6.0 multiverse" — sudo tee /etc/apt/sources.list.d/ mongodb-org-6.0.list
$ sudo apt update
$ sudo apt install -y mongodb-org
$ sudo systemctl start mongod (if '/usr/bin/mongod' is not running)
$ sudo systemctl enable mongod (ensure to automatically start it on system
boot)
2) instalar core open5gs
$ sudo add-apt-repository ppa:open5gs/latest
$ sudo apt update
$ sudo apt install open5gs
3) instalar wbui open5gs
$ sudo apt update
$ sudo apt install curl
$ curl -fsSL https://deb.nodesource.com/setup_18.x — sudo -E bash -
$ sudo apt install nodejs
$ curl -fsSL https://open5gs.org/open5gs/assets/webui/install — sudo -E bash
-
4) configurar open5gs.org
En /etc/open5gs/ copiar los archivos de configuracion amf.yaml y upf.yaml.
A través de webui, registrar el ue: conectarse a http://localhost:3000 y loguear-
se con una cuenta de administrador.
Username : admin
Password : 1423
Completar el IMSI, el contexto de seguridad (K, OPc, AMF), y el APN del
suscriptor.
IP del UE
5) habilitar NAT para rutear paquetes entre aplicaciones:
$ sudo sysctl -w net.ipv4.ip_forward=1
$ sudo sysctl -w net.ipv6.conf.all.forwarding=1
### Add NAT Rule
$ sudo iptables -t nat -A POSTROUTING -s 10.45.0.0/16 ! -o ogstun -j MAS-
QUERADE
$ sudo ip6tables -t nat -A POSTROUTING -s 2001:db8:cafe::/48 ! -o ogstun
-j MASQUERAD
```

B.3. Ejecución

1. Verificar que el proceso open5gs está corriendo con el comando: `ps -aux`
2. Espacio de nombres para el UE: `sudo ip netns add ue1, sudo ip netns list`
3. Inicio radio base: `sudo srsenb`
4. En otro terminal de línea de comandos, inicio UE: `sudo srsue`
5. En nuevo terminal hacer prueba de conexión UL: `sudo ip netns exec ue1 ping 10.45.0.1`
6. Prueba de conexión DL: `ping 10.45.0.2`

B.4. Pruebas de Funcionamiento srsRAN ZeroMQ Virtual Radio

B.4.1. Objetivo

Implementar en un único PC un sistema CN – gNB – UE, de quinta generación celular 5G, sobre el sistema ZeroMQ virtual radio de srsRan , para la transferencia de muestras de radio entre aplicaciones.

B.4.2. Ejecución

En el terminal donde ejecuta el gNB (si bien se ejecuta srseNB, el mismo se configura como NR, como se detalla en el manual), si la conexión con el core es exitosa, se despliegan los parámetros mostrados en la figura B.2

El parámetro “NG connection successful”, indica conexión exitosa del gNB al core y los parámetros RACH (Random Access Channel) indican que hay un ue conectado. También se indica la frecuencia asignada al DownLink (DL) y al UpLink (UL). En el terminal donde se ejecuta el ue, se observan los parámetros mostrados en la figura B.3, entre ellos se muestra el identificador de móvil para la celda C-RNTI (cell-radio network temporary id) y la respuesta del RAN en ra-rnti (random access-radio network temporary id). La conexión exitosa al core se manifiesta por el establecimiento de la sesión con la dirección ip 10.45.0.2

En las ventanas donde ejecutan el gNB y el UE, tipeando “t”, se despliegan las trazas del canal de comunicación correspondientes a la ejecución del comando ping de icmp entre el CN y el UE (Figuras B.4 y B.5).

En particular en el gNB (Figura B.4), en el DL, se indica el valor de Channel Quality Indicator (CQI) igual a 15. Esto corresponde a una modulación de canal de 64 QAM. El mcs (modulation and coding scheme) de 28, corresponden con una modulación de 64 QAM Radio access technology (RAT) correspondiente a new radio (NR), dado que la conexión implementada es 5G, con el core Open5Gs. Otros parámetros como bitrate (brate), completan la traza.

Apéndice B. Instalación de Plataforma srsRAN

```
wilder@srsue:~$ sudo srsenb
[sudo] password for wilder:
Active RF plugins: librsran_rf_zmq.so
Inactive RF plugins:
--- Software Radio Systems LTE eNodeB ---

Couldn't open , trying /root/.config/srsran/enb.conf
Reading configuration file /root/.config/srsran/enb.conf...
Couldn't open srb.conf, trying /root/.config/srsran/srb.conf
Couldn't open rr.conf, trying /root/.config/srsran/rr.conf
Couldn't open rb.conf, trying /root/.config/srsran/rb.conf

Built in Release mode using commit 254cc719a on branch master.

Opening 1 channels in RF device=zmq with args=fail_on_disconnect=true,tx_port=tcp://*:2000,rx_port=tcp://localhost:200
1,id=enb,base_rate=11.52e6
Supported RF device list: zmq file
CHx base_rate=11.52e6
CHx id=enb
Current sample rate is 1.92 MHz with a base rate of 11.52 MHz (x6 decimation)
CH0 rx_port=tcp://localhost:2001
CH0 tx_port=tcp://*:2000
CH0 fail_on_disconnect=true
NG connection successful
Current sample rate is 11.52 MHz with a base rate of 11.52 MHz (x1 decimation)
Current sample rate is 11.52 MHz with a base rate of 11.52 MHz (x1 decimation)
Setting frequency: DL=1842.5 Mhz, DL_SSB=1842.05 Mhz (SSB-ARFCN=368410), UL=1747.5 Mhz for cc_idx=0 nof_prb=52

=== eNodeB started ===
Type <t> to view trace
RACH: slot=171, cc=0, preamble=0, offset=0, temp_crnti=0x4601
```

Figura B.2: gNB conectado al core.

```
pablo@pablo-Precision-M4800: ~/openairinterface5g/cmake_targets/ran_build/build
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 465/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 471/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 478/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 484/2/0
[NR_PHY] =====
[NR_PHY] =====
[NR_PHY] Harq round stats for Downlink: 490/2/0
[NR_PHY] =====
```

Figura B.3: ue conectado al core.

En la traza del UE, se destacan el physical cell identifier (PCI), o sea el identificador de celda y el reference signal received power(RSRP), la potencia promedio recibida (figura B.5), además de parámetros relacionados con bit rate. Si bien es un ambiente simulado, muchos de estos parámetros están predefinidos a nivel de configuración, podemos aumentar el tráfico entre el UE y el CN para ver cómo cambian los valores de las trazas. Tener en cuenta que las figuras B.4 y B.5 se refieren simplemente a la ejecución de un ping icmp entre CN y UE. La siguiente prueba consiste en generar a través de iperf tráfico UDP de 10 Mbps durante 60 segundos, y comparar el cambio en las trazas con las generadas con icmp. Para ello configuramos el iperf del lado del CN en modo server con el siguiente comando:

- iperf3 -s -i 1

Para generar el tráfico ejecutamos en modo cliente desde el espacio de nombres

B.4. Pruebas de Funcionamiento srsRAN ZeroMQ Virtual Radio

DL										UL								
rat	pci	rnti	cqi	ri	mcs	brate	ok	nok	(%)	pusch	pucch	phr	mcs	brate	ok	nok	(%)	bsr
nr	0	4601	15	0	28	113k	3	0	0%	64.1	99.9	0	28	327k	8	0	0%	0.0
nr	0	4601	15	0	27	111k	3	0	0%	62.1	99.9	0	28	287k	7	0	0%	0.0
nr	0	4601	15	0	28	112k	3	0	0%	63.0	99.9	0	28	285k	7	0	0%	0.0
nr	0	4601	15	0	28	113k	3	0	0%	64.5	99.9	0	28	246k	6	0	0%	0.0
nr	0	4601	15	0	28	115k	3	0	0%	62.8	99.9	0	28	334k	8	0	0%	0.0
nr	0	4601	15	0	28	113k	3	0	0%	62.6	99.9	0	28	327k	8	0	0%	0.0
nr	0	4601	15	0	28	118k	3	0	0%	62.7	99.9	0	28	341k	8	0	0%	0.0
nr	0	4601	15	0	28	113k	3	0	0%	61.8	99.9	0	28	287k	7	0	0%	0.0
nr	0	4601	15	0	28	113k	3	0	0%	64.0	99.9	0	28	329k	8	0	0%	0.0
nr	0	4601	15	0	28	151k	4	0	0%	62.4	99.9	0	28	411k	10	0	0%	0.0
nr	0	4601	15	0	28	112k	3	0	0%	62.2	99.9	0	28	326k	8	0	0%	0.0

Figura B.4: gNB traza.

Signal					DL						UL			
rat	pci	rsrp	pl	cfo	mcs	snr	iter	brate	bler	ta_us	mcs	buff	brate	bler
nr	500	83	0	-13u	28	169	1.2	114k	0%	0.0	28	0.0	332k	0%
nr	500	83	0	-15u	28	169	1.2	114k	0%	0.0	28	0.0	288k	0%
nr	500	82	0	-15u	28	169	1.2	113k	0%	0.0	28	0.0	286k	0%
nr	500	83	0	-16u	28	169	1.2	113k	0%	0.0	28	0.0	287k	0%
nr	500	83	0	-14u	28	169	1.2	113k	0%	0.0	28	0.0	288k	0%
nr	500	82	0	-12u	27	169	1.1	110k	0%	0.0	28	0.0	326k	0%
nr	500	83	0	-15u	28	n/a	1.2	113k	0%	0.0	28	0.0	328k	0%
nr	500	83	0	-17u	28	169	1.2	115k	0%	0.0	28	0.0	335k	0%
nr	500	83	0	-15u	28	169	1.2	114k	0%	0.0	28	0.0	331k	0%
nr	500	83	0	-14u	28	n/a	1.2	116k	0%	0.0	28	0.0	336k	0%
nr	500	82	0	-12u	28	169	1.2	117k	0%	0.0	28	0.0	296k	0%

Signal					DL						UL			
rat	pci	rsrp	pl	cfo	mcs	snr	iter	brate	bler	ta_us	mcs	buff	brate	bler
nr	500	82	0	-12u	28	169	1.2	155k	0%	0.0	28	0.0	464k	0%
nr	500	83	0	-15u	28	n/a	1.2	112k	0%	0.0	28	0.0	326k	0%
nr	500	82	0	-15u	28	169	1.2	115k	0%	0.0	28	0.0	292k	0%
nr	500	82	0	-14u	28	169	1.2	113k	0%	0.0	28	0.0	328k	0%
nr	500	83	0	-14u	28	169	1.2	115k	0%	0.0	28	0.0	334k	0%
nr	500	83	0	-13u	28	n/a	1.2	112k	0%	0.0	28	0.0	325k	0%
nr	500	83	0	-14u	28	n/a	1.2	113k	0%	0.0	28	0.0	288k	0%
nr	500	82	0	-18u	28	169	1.2	114k	0%	0.0	28	0.0	332k	0%
nr	500	83	0	-14u	28	169	1.2	115k	0%	0.0	28	0.0	333k	0%
nr	500	83	0	-15u	28	169	1.2	114k	0%	0.0	28	0.0	289k	0%
nr	500	82	0	-12u	28	169	1.2	113k	0%	0.0	28	0.0	328k	0%

Figura B.5: UE traza.

del UE el siguiente comando, siempre con el modo server corriendo:

- `sudo ip netns exec ue1 iperf3 -c 10.45.0.1 -b 10M -i 1 -t 60`

En figuras B.6 y B.7 se despliegan las trazas asociadas a la transferencia de datos con iperf, se muestra por el lado del ue una notoria disminución de la potencia recibida a través del parámetro rsrp y del cambio en la codificación de canal DL, mcs pasa de 28 a 27 la mayor parte del tiempo. Además, como es obvio del incremento en el bit rate. Por otro lado, en el gNB queda en mayor evidencia el cambio en la codificación del canal hacia el final de la transferencia, donde el MCS pasa de 27 a quedar definitivamente en 28.

B.4.3. Conclusión

Siguiendo los tutoriales de srsRAN, se pudo implementar exitosamente el sistema ZeroMQ virtual radio, para pruebas de la plataforma con el core Open5GS. El

Apéndice B. Instalación de Plataforma srsRAN

Signal										DL				UL			
rat	pci	rsrp	pl	cfo	mcs	snr	iter	brate	bler	ta_us	mcs	buff	brate	bler			
nr	500	35	0	-17u	27	70	1.1	7.5M	0%	0.0	28	0.0	26M	0%			
nr	500	28	0	-18u	27	72	1.1	10M	0%	0.0	28	109k	29M	0%			
nr	500	24	0	-11u	27	71	1.1	14M	0%	0.0	28	120k	38M	0%			
nr	500	24	0	-6.3u	27	73	1.1	15M	0%	0.0	28	120k	38M	0%			
nr	500	23	0	-15u	27	71	1.1	14M	0%	0.0	28	124k	38M	0%			
nr	500	23	0	-12u	27	72	1.1	15M	0%	0.0	28	124k	38M	0%			
nr	500	24	0	-18u	27	71	1.1	15M	0%	0.0	28	130k	38M	0%			
nr	500	23	0	-9.9u	27	72	1.1	14M	0%	0.0	28	133k	38M	0%			
nr	500	23	0	-16u	27	71	1.1	14M	0%	0.0	28	151k	38M	0%			
nr	500	22	0	-16u	27	72	1.1	15M	0%	0.0	28	191k	38M	0%			
nr	500	24	0	-7.5u	27	71	1.1	15M	0%	0.0	28	223k	38M	0%			
Signal										DL				UL			
rat	pci	rsrp	pl	cfo	mcs	snr	iter	brate	bler	ta_us	mcs	buff	brate	bler			
nr	500	48	0	-12u	27	71	1.2	4.3M	0%	0.0	28	95k	18M	0%			
nr	500	22	0	-19u	27	72	1.1	15M	0%	0.0	28	110k	38M	0%			
nr	500	22	0	-11u	27	71	1.1	16M	0%	0.0	28	115k	38M	0%			
nr	500	24	0	-5.9u	27	72	1.1	14M	0%	0.0	28	127k	38M	0%			
nr	500	23	0	-17u	27	70	1.1	15M	0%	0.0	28	127k	38M	0%			
nr	500	22	0	-29u	27	71	1.1	14M	0%	0.0	28	120k	38M	0%			
nr	500	23	0	-18u	27	70	1.1	14M	0%	0.0	28	135k	38M	0%			
nr	500	22	0	-12u	27	71	1.1	15M	0%	0.0	28	141k	38M	0%			
nr	500	22	0	-17u	27	70	1.1	15M	0%	0.0	28	165k	38M	0%			
nr	500	23	0	-12u	27	71	1.1	14M	0%	0.0	28	205k	38M	0%			
nr	500	26	0	-13u	27	71	1.1	12M	0%	0.0	28	0.0	38M	0%			

Figura B.6: UE, traza iperf.

DL										UL								
at	pci	rnti	cqi	ri	mcs	brate	ok	nok	(%)	pusch	pucch	phr	mcs	brate	ok	nok	(%)	bsr
nr	0	4601	15	0	27	15M	112	0	0%	65.9	n/a	0	28	38M	259	0	0%	0.0
nr	0	4601	15	0	27	15M	110	0	0%	65.9	n/a	0	28	38M	259	0	0%	0.0
nr	0	4601	15	0	27	9.0M	84	0	0%	65.1	99.9	0	28	31M	260	0	0%	0.0
nr	0	4601	15	0	27	14M	106	0	0%	66.2	99.9	0	28	35M	247	0	0%	0.0
nr	0	4601	15	0	27	15M	106	0	0%	66.0	n/a	0	28	38M	244	0	0%	0.0
nr	0	4601	15	0	27	15M	107	0	0%	65.8	n/a	0	28	38M	255	0	0%	0.0
nr	0	4601	15	0	27	15M	109	0	0%	65.2	n/a	0	28	38M	254	0	0%	0.0
nr	0	4601	15	0	27	14M	107	0	0%	65.5	n/a	0	28	38M	261	0	0%	0.0
nr	0	4601	15	0	27	15M	112	0	0%	65.9	n/a	0	28	38M	254	0	0%	0.0
nr	0	4601	15	0	27	14M	105	0	0%	66.0	n/a	0	28	38M	260	0	0%	0.0
nr	0	4601	15	0	27	14M	105	0	0%	65.8	n/a	0	28	38M	265	0	0%	0.0
DL										UL								
at	pci	rnti	cqi	ri	mcs	brate	ok	nok	(%)	pusch	pucch	phr	mcs	brate	ok	nok	(%)	bsr
nr	0	4601	15	0	27	14M	107	0	0%	65.9	n/a	0	28	38M	261	0	0%	0.0
nr	0	4601	15	0	27	15M	108	0	0%	66.0	n/a	0	28	37M	255	0	0%	0.0
nr	0	4601	15	0	27	2.5M	42	0	0%	65.3	99.9	0	28	10M	153	0	0%	0.0
nr	0	4601	15	0	27	14M	105	0	0%	66.4	n/a	0	28	38M	263	0	0%	0.0
nr	0	4601	15	0	27	16M	117	0	0%	65.2	n/a	0	28	38M	258	0	0%	0.0
nr	0	4601	15	0	27	14M	108	0	0%	66.2	n/a	0	28	38M	260	0	0%	0.0
nr	0	4601	15	0	27	428k	11	0	0%	61.9	99.9	0	28	1.1M	25	0	0%	0.0
nr	0	4601	15	0	28	114k	3	0	0%	64.0	99.9	0	28	332k	8	0	0%	0.0
nr	0	4601	15	0	28	114k	3	0	0%	62.5	99.9	0	28	330k	8	0	0%	0.0
nr	0	4601	15	0	28	114k	3	0	0%	62.6	99.9	0	28	330k	8	0	0%	0.0
nr	0	4601	15	0	28	113k	3	0	0%	62.5	99.9	0	28	328k	8	0	0%	0.0

Figura B.7: gNB traza iperf.

sistema está configurado en una máquina virtual corriendo en el sistema de virtualización Hyperv de Microsoft, disponibles para pruebas de concepto y comparación con otras plataformas y / o implementaciones a lo largo del proyecto.

Apéndice C

Protocolo de Pruebas y Reporte de Resultados

C.1. Protocolo de Pruebas

Uno de los principales objetivos es evaluar la conectividad de extremo a extremo. Conectado un UE se verifica que obtiene IP, que llega con ICMP (ping) al Core Network y a una red externa. Se verifica con el comando traceroute que el camino efectivamente es utilizando los servicios de la red RAN.

Para las pruebas de navegación, la PC del Core se conecta a Internet, y configurando manualmente el DNS de la PC del UE, se verifica que llega icmp a direcciones públicas y que se puede navegar con un explorador de internet. Se valida apagando el UE, que su PC queda sin conexión.

Prueba de velocidad de Transferencia

Dada una configuración a probar, se instala un servidor iperf3 en un nodo externo conectado una red vinculada al host del core, y se prueba en forma independiente el uplink y el downlink. Se diseña un protocolo de pruebas de 2 set, por un lado se evalúa la performance de un UE conectado a un gNB y por otro la de 2 UE con y sin ruido, conectados a un gNB. Estas pruebas sirven de base para verificar el comportamiento de los algoritmos de scheduling PF y MaxTBS.

Primer set de pruebas gNB + 1 UE

Aire, gNB x310 + 1 UE x310:

- 24 PRB - 10 MHz
- 106 PRB - 40 MHz
- 217 PRB - 80 MHz

Conexión de los SDR por coaxial:

Apéndice C. Protocolo de Pruebas y Reporte de Resultados

- 217 PRB - 80 MHz
- 106 PRB - 40 MHz

RFsim:

- 24 PRB - 10 MHz
- 106 PRB - 40 MHz
- 162 PRB - 60 MHz
- 217 PRB - 80 MHz
- 273 PRB - 100 MHz

Segundo set de pruebas gNB + 2 UE

Aire, 24PRB - 10 MHz - gNB x310 + 2UE B200:

- Caso 1 PF (Proportional Fair)
- Caso 2 MaxTBS (Maximize Transport Block Size)

RFsim 106PRB - 40 MHz:

- Caso 1 PF
- Caso 2 MaxTBS
- Caso 3 MaxTBS con ruido

C.2. Reporte de resultados

Primer set de pruebas gNB + 1 UE

En las tablas y figuras C.1 y C.2, se muestran los resultados del primer set de pruebas, el tope en los iperf es para estabilizar el sistema y no generar retransmisiones innecesarias, dado que la finalidad es medir el máximo *throughput* (*Th*). También se agregaron los máximos teóricos dados por la ecuación: C.1 y los cálculos para el *throughput* de DL y UL, dados por las ecuaciones C.2 y C.3. En la tabla C.6 están los máximos rendimientos teóricos para todos los PRB usados en las pruebas.

$$Th(Mb/s) = 10^{-6} * \sum_{j=1}^j (v_{layers}^{(j)} * Q_m^{(j)} * f^{(j)} * R_{max} * \frac{N_{PRB}^{BW(j),\mu} * 12}{T_s^\mu} * (1 - OH^{(j)})) \quad (C.1)$$

Primer Set de Pruebas, 1 UE x310					
Medio	DL/DL	PRB	Throughput(Mbps)	Banda	Tope iperf(Mbps)
Aire	DL	24	7.73	n41	20
Aire	UL	24	4.7	n41	5
Aire	DL	106	42.8	n41	50
Aire	UL	106	8.21	n41	20
Aire	DL	217	63.7	n78	sin tope
Aire	UL	217	12.5	n78	20
Coaxial	DL	106	19	n41	80
Coaxial	UL	106	8.18	n41	20
Coaxial	DL	217	75.1	n78	80
Coaxial	UL	217	8.26	n78	20
Teórico	DL	24	28.616	n41	80
Teórico	UL	24	8.831	n41	20
Teórico	DL	106	126.389	n78	80
Teórico	UL	106	39.002	n78	20
Teórico	DL	217	258.739	n78	80
Teórico	UL	217	79.844	n78	20

Tabla C.1: Tabla Primer Set de Pruebas, en aire y cable coaxial. Se agregan valores teóricos para cada número de PRB.

- $j = 1$
- $v_{layers}^{(j)} = 1$
- $Q_m^{(j)} = 6$
- $f^{(j)} = 1$
- $R_{max} \simeq 0,926$
- $N_{PRB}^{BW(j),\mu} = 217$
- $\mu = 1$
- $T_s^{\mu=1} \simeq 3,57 \times 10^{-5}$
- $OH^{(j)} = 0,14 DL; 0,08 UL$

Factor de ajuste según formato de la trama:

- $DL = \frac{7 + \frac{6}{14}}{10} \simeq 0,743$

Apéndice C. Protocolo de Pruebas y Reporte de Resultados

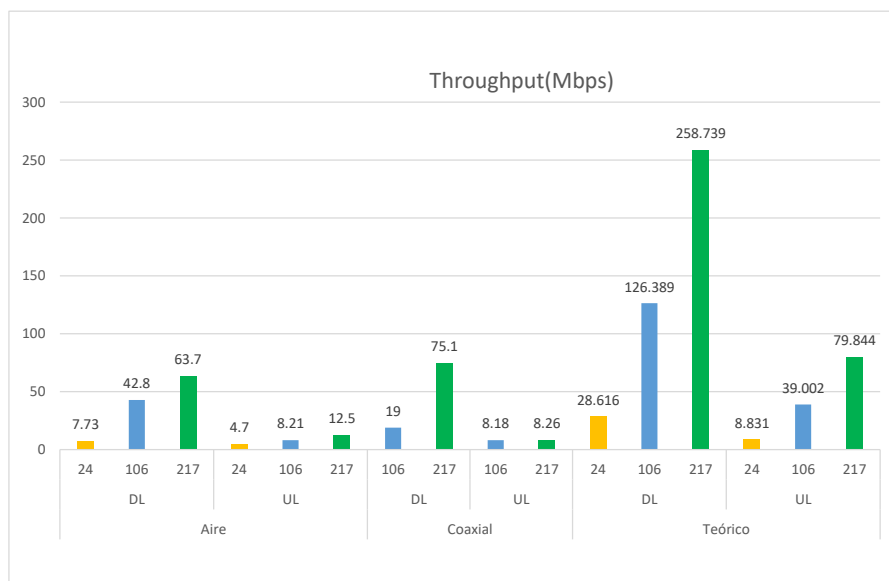


Figura C.1: Throughput según medio y PRB del primer set de pruebas, se contrasta con datos máximos teóricos

$$\blacksquare UL = \frac{2 + \frac{4}{14}}{10} \simeq 0,214$$

$$Th_{DL}(Mb/s) = 10^{-6}(1 \times 6 \times 1 \times 0,926 \times \frac{217 \times 12}{3,57 \times 10^{-5}} \times (1 - 0,14)) \times 0,743 \quad (C.2)$$

$$Th_{DL}(Mb/s) = 258,74$$

$$Th_{UL}(Mb/s) = 10^{-6}(1 \times 6 \times 1 \times 0,926 \times \frac{217 \times 12}{3,57 \times 10^{-5}} \times (1 - 0,08)) \times 0,214 \quad (C.3)$$

$$Th_{UL}(Mb/s) = 79,84$$

Segundo set de pruebas gNB + 2 UE

En la tabla y figura C.3 , se muestra el trthroughput en el caso con 2 UE B200. Se describen los casos con schedule Proportional Fair (PF) y Maximize transport block size (MaxTBS). Además se hacen pruebas de iperf con un servidor externo a la red OAI y pruebas con servidor iperf configurado en el UE2, o sea dentro de la red OAI.

En la tabla y figura C.4, se representa el trthroughput de 2 UE en RFSim, configurados con 106 PRB en un canal sin ruido. Se los muestra transmitiendo

Primer Set de Pruebas, RFsim					
Medio	DL/DL	PRB	Throughput(Mbps)	Banda	Tope iperf(Mbps)
RFsim	DL	24	133	n41	150
RFsim	UL	24	40.7	n41	100
RFsim	DL	106	248	n78	260
RFsim	UL	106	83.3	n78	50
RFsim	DL	162	279	n78	300
RFsim	UL	162	99.6	n78	150
RFsim	DL	217	180	n78	250
RFsim	UL	217	51.0	n78	80
RFsim	DL	273	190	n77	250
RFsim	UL	273	53.9	n77	70

Tabla C.2: Tabla Primer Set de Pruebas, RFsim.

Segundo Set de Pruebas, 2 UE B200						
Medio	DL/DL	PRB	Th (Mbps)	Banda	tipo schedule	server iperf
Aire	DL	24	5.63	n41	PF	externo
Aire	UL	24	0.87	n41	PF	externo
Aire	DL	24	4.67	n41	MaxTBS	externo
Aire	UL	24	0.86	n41	MaxTBS	externo
Aire	DL	24	1.29	n41	PF	UE2
Aire	UL	24	1.73	n41	PF	UE2
Aire	DL	24	3.88	n41	MaxTBS	UE2
Aire	UL	24	0.75	n41	MaxTBS	UE2

Tabla C.3: Tabla Segundo Set de Pruebas, 2 UE B200.

por separado y en simultáneo. En ambos casos se compara performance con los schedule PF y MaxTBS.

En la tabla y figura C.5, se expone el trthroughput asociado al MaxTBS, con los UE con ruido, conectados de a uno, en forma simultanea y alternado en uno y otro la presencia de ruido en el canal

Apéndice C. Protocolo de Pruebas y Reporte de Resultados

Segundo Set de Pruebas, 2 UE RFsim SIN Ruido						
UEs simultáneos	DL/DL	PRB	Th (Mbps)	Banda	tipo schedule	UE
1	UL	106	18.7	n78	PF	UE1
1	UL	106	15.9	n78	PF	UE2
1	DL	106	141	n78	PF	UE1
1	DL	106	149	n78	PF	UE2
2	DL	106	86.8	n78	PF	UE1
2	DL	106	84.0	n78	PF	UE2
2	UL	106	9.15	n78	PF	UE1
2	UL	106	8.78	n78	PF	UE2
1	UL	106	18.2	n78	MaxTBS	UE1
1	UL	106	15.7	n78	MaxTBS	UE2
1	DL	106	138	n78	MaxTBS	UE1
1	DL	106	140	n78	MaxTBS	UE2
2	DL	106	57.9	n78	MaxTBS	UE1
2	DL	106	112	n78	MaxTBS	UE2
2	UL	106	4.21	n78	MaxTBS	UE1
2	UL	106	15.7	n78	MaxTBS	UE2

Tabla C.4: Tabla Segundo Set de Pruebas, RFsim Sin Ruido.

Segundo Set de Pruebas, 2 UE RFsim Con Ruido					
UEs simultáneos	DL/DL	PRB	Th (Mbps)	tipo schedule	UE
1	UL	106	10.6	MaxTBS	UE1 ruido
1	UL	106	9.83	MaxTBS	UE2 ruido
2	UL	106	8.97	MaxTBS	UE1
2	UL	106	2.94	MaxTBS	UE2 ruido
2	DL	106	78.3	MaxTBS	UE1
2	DL	106	12.1	MaxTBS	UE2 ruido
2	DL	106	11.3	MaxTBS	UE1 ruido
2	DL	106	79.3	MaxTBS	UE2

Tabla C.5: Tabla Segundo Set de Pruebas, MaxTBS con Ruido.

C.2. Reporte de resultados

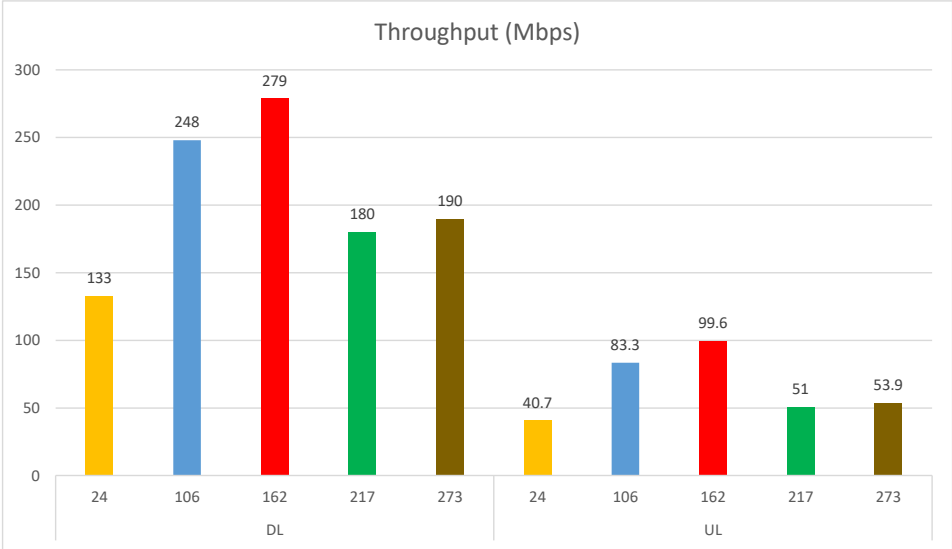


Figura C.2: Throughput en RFSim Según PRB.

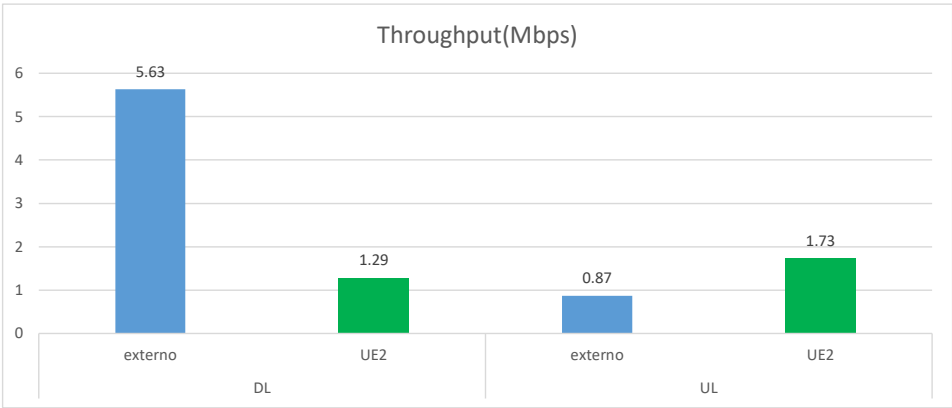


Figura C.3: Throughput en Aire Según ubicación de servidor iperf.

Apéndice C. Protocolo de Pruebas y Reporte de Resultados

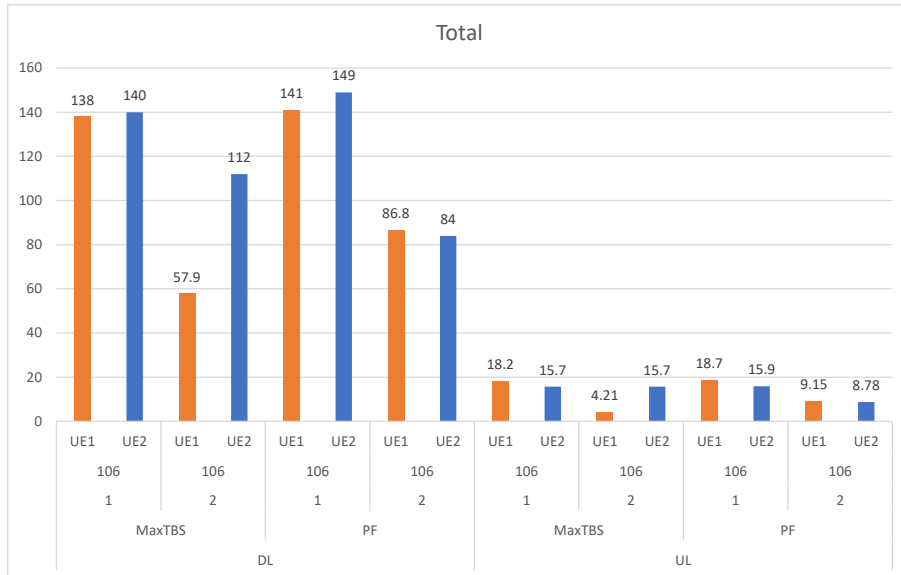


Figura C.4: Throughput en RFSim Según tipo de Algoritmos de scheduler.

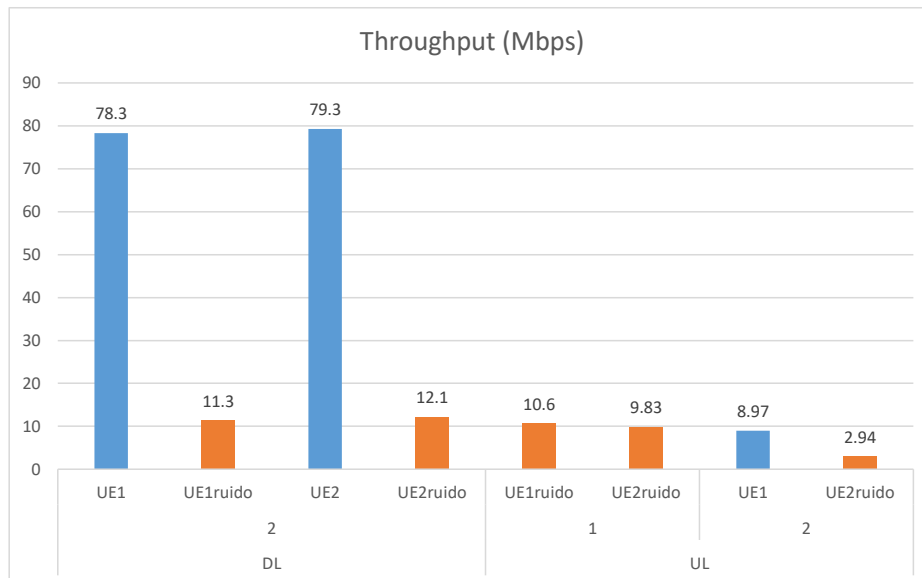


Figura C.5: Throughput en RFSim, con MaxTBS y ruido.

C.2. Reporte de resultados

Throughput teórico 1 layer 64 QAM maximo		
DL/DL	PRB	Throughput(Mbps)
DL	24	28.616
UL	24	8.831
DL	106	126.389
UL	106	39.002
DL	162	193.160
UL	162	59.607
DL	217	258.739
UL	217	79.844
DL	273	325.511
UL	273	100.448

Tabla C.6: Tabla de throughput teórico según la fórmula presentada en C.1.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice D

Scheduler Alternativo

Siguiendo el estilo OAI, la documentación básica del software está en el archivo `readme`. Allí se esquematiza su funcionamiento y se dan las instrucciones para activarlo. En la sección siguiente se detalla el software modificado.

D.1. Readme

Esta versión del software incluye formas alternativas de scheduling al algoritmo previsto por OAI.

La versión original ordena los UE por un coeficiente de la forma $\frac{TBS}{throughput}$ en cada iteración por subtrama. Luego se le asignan todos los recursos al primer UE de la lista. Si se terminan sus datos en buffer, se le asignan al siguiente. Procesos iguales ocurren en UL y DL.

En esta versión el coeficiente es $\frac{TBS^\alpha}{throughput^\beta}$. Se agrega un criterio aleatorio de ordenado entre los UE con igual coeficiente. La lista así ordenada es utilizada por las funciones originales para asignar recursos.

Configuración

En el archivo de configuración del gNB se pueden incluir cuatro parámetros en la sección MACRLCs:

- `ul.alfa = 1;#` por defecto 1, exponente del tbs en el numerador
- `ul.beta = 1;#` por defecto 1, exponente del throughput en el denominador
- `dl.alfa = 1;#` por defecto 1, exponente del tbs en el numerador
- `dl.beta = 1;#` por defecto 1, exponente del throughput en el denominador

Cuando se omite la declaración toman el valor 1.

Apéndice D. Scheduler Alternativo

Opciones de funcionamiento

Si en uno de los procesos de *scheduling*, UL o DL, $\alpha = 1$ y $\beta = 1$, el sistema funciona con *Proportional Fair* sin ponderación. El proceso de ordenado aleatorio entre coeficientes iguales es irrelevante, porque en PF, el que es adjudicado cambia su coeficiente como resultado de la asignación.

Si $\alpha = 1$ y $\beta = 0$, funciona con el criterio MaxTBS. Se asigna al de mayor TBS, y si hay varios con igual TBS se ordena en forma aleatoria entre ellos.

Si $\alpha = 0$ y $\beta = 0$ todos los UE se ordenan en forma aleatoria.

Si $\alpha = 0$ y $\beta = 1$ se le da máxima prioridad a quien ha traficado menos, con independencia de sus condiciones de enlace.

Otros valores enteros de α y β son posibles.

D.2. Modificaciones al Software

Criterios generales

El sistema que se está modificando es grande y complejo. La mayoría de los estados y variables se guardan en variables globales de donde son consultadas, copiadas y vueltas a salvar desde distintos procesos, por lo que se debe preservar todos las acciones existentes, al menos hasta tener la certeza completa de su uso. A modo de ejemplo, el cálculo del throughput de un UE hecho para el coeficiente de ordenado del scheduling, es el valor almacenado como el KPI correspondiente a ese UE. Reemplazar la función de preproceso completa implica rehacer los cálculos aunque no se usen, para mantener los datos generados.

Por otro lado, es un sistema con funciones de control de tiempo sobre los procesos principales. Las cuentas se hacen en enteros, con rotaciones de bit para divisiones entre 2^n , las listas se hacen con arrays de punteros con tope, etc.

Para no alterar funciones no estudiadas, y tener impactos mínimos en los tiempos de ejecución, se buscó hacer modificaciones mínimas para lograr el objetivo a partir del estudio del tipo de ingeniería inversa sobre la solución implementada originalmente.

Definición y lectura de parámetros de configuración

1. Se incluyen los exponentes de ponderación a ser configurados como elementos del struct `gNB_MAC_INST_s` que define al tipo `gNB_MAC_INST` definido en el módulo:

```
openairinterface5g/openair2/LAYER2/NR_MAC_gNB/nr_mac_gNB.h
```

```

.

    ///COEFICIENTES SCHEDULER MAQ5G
    uint32_t dl_alfa;
    uint32_t dl_beta;
    uint32_t ul_alfa;
    uint32_t ul_beta;
```

2. Se modifica el módulo

openairinterface5g/openair2/GNB_APP/MACRLC_nr_paramdef.h

para incluir en la lista de parámetros posibles de leer de la sección MACRLCs del archivo de configuración. En esta lista se incluye el nombre del parámetro a buscar en el archivo, el tipo del dato, su valor por defecto, un texto de ayuda, y si hay condiciones de verificación y banderas de condiciones.

Se puede profundizar en los documentos incluidos en:

openairinterface5g/common/config/DOC/config/devusage.md

Esta lista de parámetros es leída del archivo de configuración al inicio del programa, utilizando la librería libconfig. La configuración se hace en la función void RCconfig_nr_macrlc()

3. Se modifica la función la función void RCconfig_nr_macrlc() del módulo

openairinterface5g/openair2/GNB_APP/gnb_config.c

para guardar en las variables globales de RC.nrmac del tipo gNB_MAC_INST los parámetros leídos en la configuración.

```
RC.nrmac[j]->dl_alfa = *(MacRLC_ParamList.paramarray[j] [MACRLC_EXPO_DL_ALFA].uptr);
RC.nrmac[j]->dl_beta = *(MacRLC_ParamList.paramarray[j] [MACRLC_EXPO_DL_BETA].uptr);
RC.nrmac[j]->ul_alfa = *(MacRLC_ParamList.paramarray[j] [MACRLC_EXPO_UL_ALFA].uptr);
RC.nrmac[j]->ul_beta = *(MacRLC_ParamList.paramarray[j] [MACRLC_EXPO_UL_BETA].uptr);
```

Se escribe en el log de inicio los valores leídos de estas variables.

Modificación del *scheduler*

1. Modificación del preproceso en DL hecho en la función pf_dl en el módulo

.../openair2/LAYER2/NR_MAC_gNB/gNB_scheduler_dlsch.c

Se crean variables locales para los exponentes y se copian los valores globales en ellas.

```
//Exponentes para priorizar factores de PF en DL.
// exponente de TBS en numerador
const float alfa_DL = RC.nrmac[0]->dl_alfa;
// exponente de throughput en divisor
const float beta_DL = RC.nrmac[0]->dl_beta;
```

Se modifica el cálculo del coeficiente con las potencias.

```
float coeff_ue = pow((float) tbs, alfa_DL) / pow(UE->dl_thr_ue,beta_DL);
```

Apéndice D. Scheduler Alternativo

Esta cuenta se hace en punto flotante para mantener la capacidad de discriminar diferencias. Ya es así en el algoritmo original.

En el mismo módulo, pero fuera de la función `pf_dl` se modifica el tipo `UESched_t` que define los nodos del array que lista los UE a ser agendados, para incluir un campo entero donde guardar el valor aleatorio del desempate.

```
typedef struct UESched_s {
    float coef;
    NR_UE_info_t * UE;
    int desempate;
} UESched_t;
```

Luego se agrega ese valor entero pseudoaleatorio de desempate para cada UE de la lista, en cada llamado a la función.

```
/* Create UE_sched list for UEs eligible for new transmission*/
UE_sched[curUE].coef=coeff_ue;
UE_sched[curUE].UE=UE;
//se agrega criterio aleatorio de desempate en cada iteración
UE_sched[curUE].desempate=rand();
```

2. Se modifica la función de comparación utilizada por la función `qsort`. La original devuelve la comparación entre coeficientes, esta también si no son iguales, pero si son iguales devuelve la comparación entre los valores de desempate.

```
static int comparator(const void *p, const void *q) {
    if (((UESched_t*)p)->coef != ((UESched_t*)q)->coef){
        return ((UESched_t*)p)->coef < ((UESched_t*)q)->coef;
    } else {
        return ((UESched_t*)p)->desempate < ((UESched_t*)q)->desempate;
    }
}
```

- 3.
4. Modificación del preproceso en UL hecho en la función `pf_ul` en el módulo

.../openair2/LAYER2/NR_MAC_gNB/gNB_scheduler_ulsch.c

Las modificaciones son iguales a las hechas en DL.

Se definen las variables:

```
//Exponentes para ajustar el algoritmo PF. mac5g
/exponente para TBS, nuevo criterio de desicion poniendo en 1
const float alfa_UL = RC.nrmac[0]->ul_alfa; /
// exponente para throughput previo, se lo elimina de la decisión poniendo 0.
const float beta_UL = RC.nrmac[0]->ul_beta;
```

Se modifica el cálculo del coeficiente con las potencias.

```
// coeficiente modificado
float coeff_ue = pow((float) tbs, alfa_UL) / pow(UE->ul_thr_ue, beta_UL);
```


D.2. Modificaciones al Software

En el mismo módulo, pero fuera de la función `pf_ul` se modifica el tipo `UESched_t` que define los nodos del array que lista los UE a ser agendados, para incluir un campo entero donde guardar el valor aleatorio del desempate.

```
typedef struct UESched_s {
    float coef;
    NR_UE_info_t * UE;
    int desempate;//para aleatorizar la seleccion de 2 ue con igual TBS
} UESched_t;
```

Luego se agrega ese valor entero pseudoaleatorio de desempate para cada UE de la lista, en cada llamado a la función.

```
UE_sched[curUE].coef=coeff_ue;
UE_sched[curUE].UE=UE;
UE_sched[curUE].desempate=rand();
```

5. Se modifica la función de comparación utilizada por la función `qsort`. La función es idéntica a la utilizada en DL.

```
static int comparator(const void *p, const void *q) {
    if (((UESched_t*)p)->coef != ((UESched_t*)q)->coef){
        return ((UESched_t*)p)->coef < ((UESched_t*)q)->coef;
    } else {
        return ((UESched_t*)p)->desempate < ((UESched_t*)q)->desempate;
    }
}
```

Exportación de datos

Para poder evaluar el funcionamiento del sistema se exporta a archivos `.csv` los datos de asignación de cada slot en que interviene el scheduler. Son dos archivos, uno de UL y otro de DL. Cada fila tiene la información de un UE con asignación en un slot. Si en un slot no hay asignaciones, no se genera la fila. Si dos o más UE reciben recursos en el mismo slot, habrá una fila para cada uno. Las columnas del archivo `DL_planilla.csv` contienen:

- Frame - Número de trama (0 a 1023)
- Slot - Número de slot en la trama (0 a 19)
- RNTI - id temporal del UE
- DCI L
- start - Posición del primer RB asignado
- RBs - Cantidad de RB asignados
- startSymbol - primer símbolo OFDM asignado en el slot
- nb_symbol Cantidad de símbolos OFDM asignados en el slot
- dmrspos

Apéndice D. Scheduler Alternativo

- MCS Esquema de modulación y codificación de ese UE en ese slot.
- nrOfLayers - Número de capas en uso en ese slot por ese UE.
- TBS Tamaño del bloque de transporte asignado en Bytes.
- HARQ PID
- round
- RV
- NDI
- dl_data_to_ULACK
- UL frame
- UL slot
- PUCCH allocation
- TPC

Las columnas del archivo UL_planilla.csv contienen:

- Frame - Número de trama (0 a 1023)
- Slot - Número de slot en la trama (0 a 19)
- RNTI - id temporal del UE
- UL sched-frame Trama para la que recibe agenda
- UL sched-slot Slot para el que recibe agenda
- DCI L
- start - Posición del primer RB asignado
- RBs - Cantidad de RB asignados
- startSymbol - primer símbolo OFDM asignado en el slot
- nb_symbol Cantidad de símbolos OFDM asignados en el slot
- dmrspos
- MCS Table Tabla que debe usar para interpretar el MCS
- MCS Esquema de modulación y codificación de ese UE en ese slot.
- nrOfLayers - Número de capas en uso en ese slot por ese UE.
- num dmrs cdm gprs no data

- TBS Tamaño del bloque de transporte asignado en Bytes.
- HARQ PID
- round
- RV *redundancy version*
- NDI
- est
- est BSR
- TPC

Se comentaron los parámetros utilizados en el análisis.
Las modificaciones para generar estos archivos son:

1. Se crean las planillas y los encabezados en el main() del programa ubicado en el módulo

openairinterface5g/executables/nr – softmodem.c

```
//lineas agregadas para tirar log a archivo csv MAQ5G
//*****
//*****
FILE *DL_planilla1;
DL_planilla1 = fopen("DL_planilla.csv","w");
fprintf(DL_planilla1, "Frame,Slot,RNTI,UL sched-frame,UL sched-slot,DCI L,RB Start,
RB size,Symbol start,Symbol Size, DMRS pos,MCS Table,MCS,nrOfLayers,
num_dmrs_cdm_gprs_no_data,TBS,HARQ PID,round,RV,NDI,est,est BSR,TPC\n");
fclose(DL_planilla1);

FILE *UL_planilla1;
UL_planilla1 = fopen("UL_planilla.csv","w");
fprintf(UL_planilla1, "Frame,Slot,RNTI,DCI L,RB Start,RB size,Symbol start,Symbol
Size,DMRS pos,MCS,nrOfLayers,TBS,HARQ PID,round,RV,NDI,dl_data_to_ULACK,UL frame,
UL slot,PUCCH allocation,TPC\n");
fclose(UL_planilla1);
```

2. En cada iteración se escribe una línea de cada planilla. En la función void nr_schedule_ue_spec que implementa el armado de los canales de transporte y se los entrega a la capa 1, ubicada en el módulo

.../openair2/LAYER2/NR_MAC_gNB/gNB_scheduler_dlsch.c

Se abre el archivo, y se agrega una fila en cada posible UE al recorrer la lista ejecutando asignaciones de recursos.

```
FILE *DL_planilla;
DL_planilla = fopen("DL_planilla.csv","a");
.
.
.
```

Apéndice D. Scheduler Alternativo

```
/* Se agrega un fprintf para visualizar los parámetros en uso*/
fprintf(DL_planilla,
        "%04x,%d,%3d,%3d,%2d,%2d,%x,%2d,%d,%4d,%2d,%d,%d,%d,%d,%d,%d\n",
        frame,
        slot,
        rnti,
        sched_ctrl->aggregation_level,
        sched_pdsch->rbStart,
        sched_pdsch->rbSize,
        tda_info->startSymbolIndex,
        tda_info->nrOfSymbols,
        dmrs_params->dl_dmrs_symb_pos,
        sched_pdsch->mcs,
        nrOfLayers,
        TBS,
        current_harq_pid,
        harq->round,
        nr_rv_round_map[harq->round%4],
        harq->ndi,
        pucch->timing_indicator,
        pucch->frame,
        pucch->ul_slot,
        sched_pdsch->pucch_allocation,
        sched_ctrl->tpc1);
//*****
.
.
.
fclose(DL_planilla);
```

3. El mismo proceso de apertura y cierre de archivos y agregado de líneas ocurre en UL en la función void nr_schedule_ulsch del módulo ya revisado

.../openair2/LAYER2/NR_MAC_gNB/gNB_scheduler_ulsch.c

D.3. Análisis de Planillas

Las planillas generadas son muy grandes. Solo suman filas cuando hay tráfico, pero en esos períodos agregan dosmil filas por segundo. Se revisaron utilizando Paython en un Jupyter Notebook. Lo primero es agregar una estampa de tiempo a cada línea. Para eso se asigna un número de secuencia que se inicia en cero y se incrementa cada vez que reinicia la secuencia de tramas. Luego se obtiene el tiempo desde el origen en ms sumando secuencia, trama y slot, por sus periodos y restando los valores iniciales.

$$t = (N \times 1024 + F) \times 10 + s \times 0,5 - F_0 \times 10 - s_0 \times 0,5$$

donde N es la cantidad de veces que se reinició la secuencia de tramas, F es el número de trama, s el número de slot en esa trama y F_0 y s_0 son los valores de la primer trama y el primer slot de la planilla.

Usando esa base de tiempos, se acumularon los bytes asignados a cada UE(dato de la columna TBS) por segundo, para representarlos en una escala apreciable.

Apéndice E

Bandas de Frecuencia

Se presentan las bandas de frecuencia habilitadas para NR en la recomendación TS 38.104 capítulo 5 [6].

Banda	Modo	f (MHz)	UL	DL	Ancho de canal (MHz)
n1	FDD	2100	1920 – 1980	2110 – 2170	5, 10, 15, 20, 25, 30, 40, 45, 50
n2	FDD	1900	1850 – 1910	1930 – 1990	5, 10, 15, 20, 25, 30, 35, 40
n3	FDD	1800	1710 – 1785	1805 – 1880	5, 10, 15, 20, 25, 30, 35, 40, 45, 50
n5	FDD	850	824 – 849	869 – 894	5, 10, 15, 20, 25
n7	FDD	2600	2500 – 2570	2620 – 2690	5, 10, 15, 20, 25, 30, 35, 40, 50
n8	FDD	900	880 – 915	925 – 960	5, 10, 15, 20, 25, 35
n12	FDD	700	699 – 716	729 – 746	5, 10, 15
n13	FDD	700	777 – 787	746 – 756	5, 10
n14	FDD	700	788 – 798	758 – 768	5, 10
n18	FDD	850	815 – 830	860 – 875	5, 10, 15
n20	FDD	800	832 – 862	791 – 821	5, 10, 15, 20
n24	FDD	1600	1626.5–1660.5	1525 – 1559	5, 10
n25	FDD	1900	1850 – 1915	1930 – 1995	5, 10, 15, 20, 25, 30, 35, 40, 45
n26	FDD	850	814 – 849	859 – 894	5, 10, 15, 20, 25, 30
n28	FDD	700	703 – 748	758 – 803	5, 10, 15, 20, 25, 30
n29	SDL	700	—	717 – 728	5, 10
n30	FDD	2300	2305 – 2315	2350 – 2360	5, 10
n34	TDD	2100	2010 – 2025		5, 10, 15
n38	TDD	2600	2570 – 2620		5, 10, 15, 20, 25, 30, 40
n39	TDD	1900	1880 – 1920		5, 10, 15, 20, 25, 30, 40
n40	TDD	2300	2300 – 2400		5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
n41	TDD	2500	2496 – 2690		5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100
n46	TDD	5200	5150 – 5925		10, 20, 40, 60, 80, 100
n47	TDD	5900	5855 – 5925		10, 20, 30, 40
n48	TDD	3500	3550 – 3700		5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100
n50	TDD	1500	1432 – 1517		5, 10, 15, 20, 30, 40, 50, 60, 80
n51	TDD	1500	1427 – 1432		5
n53	TDD	2400	2483.5 – 2495		5, 10
n54	TDD	1600	1670 – 1675		5
n65	FDD	2100	1920 – 2010	2110 – 2200	5, 10, 15, 20, 50
n66	FDD	1700 2100	1710 – 1780	2110 – 2200	5, 10, 15, 20, 25, 30, 35, 40, 45
n67	SDL	700	—	738 – 758	5, 10, 15, 20

Tabla E.1: Tabla de bandas de frecuencias FR1.

Apéndice E. Bandas de Frecuencia

Banda	Modo	f (MHz)	UL	DL	Ancho de canal (MHz)
n70	FDD	2000	1695 – 1710	1995 – 2020	5, 10, 15, 20, 25
n71	FDD	600	663 – 698	617 – 652	5, 10, 15, 20, 25, 30, 35
n74	FDD	1500	1427 – 1470	1475 – 1518	5, 10, 15, 20
n75	SDL	1500	—	1432 – 1517	5, 10, 15, 20, 25, 30, 40, 50
n76	SDL	1500	—	1427 – 1432	5
n77	TDD	3700	3300 – 4200		10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
n78	TDD	3500	3300 – 3800		10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
n79	TDD	4900	4400 – 5000		10, 20, 30, 40, 50, 60, 70, 80, 90, 100
n80	SUL	1800	1710 – 1785	—	5, 10, 15, 20, 25, 30, 40
n81	SUL	900	880 – 915	—	5, 10, 15, 20
n82	SUL	800	832 – 862	—	5, 10, 15, 20
n83	SUL	700	703 – 748	—	5, 10, 15, 20, 25, 30
n84	SUL	2100	1920 – 1980	—	5, 10, 15, 20, 25, 30, 40, 50
n85	FDD	700	698 – 716	728 – 746	5, 10, 15
n86	SUL	1700	1710 – 1780	—	5, 10, 15, 20, 40
n89	SUL	850	824 – 849	—	5, 10, 15, 20, 50
n90	TDD	2500	2496 – 2690		5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100
n91	FDD	800 1500	832 – 862	1427 – 1432	5, 10
n92	FDD	800 1500	832 – 862	1432 – 1517	5, 10, 15, 20
n93	FDD	900 1500	880 – 915	1427 – 1432	5, 10
n94	FDD	900 1500	880 – 915	1432 – 1517	5, 10, 15, 20
n95	SUL	2100	2010 – 2025	—	5, 10, 15
n96	TDD	6000	5925 – 7125		20, 40, 60, 80, 100
n97	SUL	2300	2300 – 2400	—	5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
n98	SUL	1900	1880 – 1920	-	5, 10, 15, 20, 25, 30, 40
n99	SUL	1600	1626.5–1660.5	-	5, 10
n100	FDD	900	874.4 – 880	919.4 – 925	5
n101	TDD	1900	1900 – 1910		5, 10
n102	TDD	6200	5925 – 6425		20, 40, 60, 80, 100
n104	TDD	6700	6425 – 7125		20, 30, 40, 50, 60, 70, 80, 90, 100
n105	FDD	600	663 – 703	612 – 652	5, 10, 15, 20, 25, 30, 35

Tabla E.2: Tabla de bandas de frecuencias FR1, continuación.

Banda	f (GHz)	Nombre	UL / DL	Ancho del Canal (MHz)
n257	28	LMDS	26.50 – 29.50	50, 100, 200, 400
n258	26	K-band	24.25 – 27.50	50, 100, 200, 400
n259	41	V-band	39.50 – 43.50	50, 100, 200, 400
n260	39	Ka-band	37.00 – 40.00	50, 100, 200, 400
n261	28	Ka-band	27.50 – 28.35	50, 100, 200, 400
n262	47	V-band	47.20 – 48.20	50, 100, 200, 400
n263	60	V-band	57.00 – 71.00	100, 400, 800, 1600, 2000

Tabla E.3: Tabla de bandas de frecuencias FR2.

Apéndice F

Lecciones Aprendidas

F.1. Configuración de teléfonos comerciales

Lo primero es poner los aparatos móviles en modo ingeniería, con esta herramienta es posible visualizar la potencia transmitida y recibida por el móvil, entre otros parámetros útiles para la conexión. Uno de los problemas más comunes es la saturación de potencia por parte de los móviles hacia el gNB, esto provoca que el gNB caiga abruptamente ni bien los móviles intenten conectar. Acá se inicia un proceso de ensayo y error, donde se ajusta en el archivo de configuración del gNB los parámetros de atenuación en recepción y transmisión.

Otro parámetro importante es `min rxtxtime` que tiene que ver con el número de slot de separación entre UL y DL. Habitualmente se configura con el valor de 6 en los UE implementados con SDR, pero para el caso del móvil Samsung A33 hay que configurarlo con el valor 2. Esta información fue obtenida del foro de OAI.

Para la configuración en el equipo Samsung A33:

Nombre: Un nombre, puede ser cualquiera

Apn: oai (o el que corresponda según las opciones de DNN/APN en el core)

MCC-MNC: 001-01 (Debe tomar automáticamente de lo que esté grabado en el chip)

Tipo de APN: default

Protocolo APN: IPv4

Portadora: NR

Tipo de operador de red virtual móvil: SPN (debe tomar el nombre de la red grabado en el chip)

El resto de los parámetros en "No definido"

Roaming de datos: ON

Llamadas VoLTE: OFF

Luego de la configuración del teléfono, activar modo avión y desactivar para que empiece el proceso de registro.

F.2. Correr un UE en RFSimulator con ruido blanco.

Agregar en la línea de ejecución del UE las opciones `-- rfsimulator.options chanmod -- rfsimulator.modelname AWGN -- rfsimulator.noise_power_dB -10`, agrega ruido en la recepción. También puede agregarse en el gNB.

Este comando agrega un nivel de ruido de -10 dB respecto del valor de señal de referencia. Es una prueba al tanteo. -10 es un nivel de ruido que se siente, -5 es mucho ruido, casi está cayéndose la conexión, -20 o -30 no se notan en una verificación simple. El ruido agregado se ve en la constelación recibida.

F.3. Uso de los logs

El sistema tiene la posibilidad de variar el nivel de detalle del log en consola, en la salida estándar. Eso se puede hacer en el archivo de configuración, variando el nivel de detalle de cada una de las 8 entradas indicadas:

```
log_config :
{
    global_log_level           ="info";
    hw_log_level               ="info";
    phy_log_level              ="info";
    mac_log_level              ="info";
    rlc_log_level              ="info";
    pdcp_log_level            ="info";
    rrc_log_level              ="info";
    f1ap_log_level             ="debug";
};
```

Los niveles de log pueden ser error, warn, analysis, info, debug, trace. Si se setea en un valor, se envía a la consola los mensajes de ese y todos los niveles anteriores.

El problema es que usarlo así es tan poco práctico que lo hace inútil. Primero, porque cambiar el archivo de configuración en cada uso es tedioso, y se pierde la referencia de cuál es el que se considera original. O cuál es la referencia que se usa en cada caso. Guardar copias del archivo con cada modificación termina siendo inmanejable. Para cambios de un solo uso, es mejor sobrescribir el parámetro en el comando de ejecución, agregando la opción: `-- log_config.phy_log_level debug`, donde el comando indica la categoría a modificar y el valor es el nivel al que se configura.

Esto tiene otra dificultad extra, en los niveles más detallados la cantidad de mensajes es enorme, muchos de ellos, uno por slot, 2000 líneas por segundo. Por lo tanto, verlo en vivo es imposible, y buscarlos en un archivo salvado de la salida estándar es muy difícil. Si se para la ejecución y se busca en la consola, solo se encontrarán los mensajes de los últimos instantes que completarán la salida. Lo práctico es saber de antemano que se busca, encontrar el mensaje en el código,

F.4. Otras Herramientas

y luego hacer un grep con parte del texto buscado para filtrar solo las líneas de interés. Por ejemplo, para leer directamente de la salida del gNB el offset requerido para el SSB en el UE, se puede ejecutar en una sola línea por ejemplo así:

```
sudo -E RFSIMULATOR=server ./nr-softmodem --rfsim --sa -0
~/Desktop/conf/gnb.sa.band77.fr1.273PRB.2x2.usrpn300_test.conf
--log_config.phy_log_level debug | grep "SSB first"
```

O para mandar a un archivo el resultado del cálculo del coeficiente en el scheduler se puede ejecutar:

```
sudo ./nr-softmodem -0 ../../../../targets/PROJECTS/GENERIC-NR-5GC/
CONF/gnb.sa.band41.fr1.106PRB.usrpb210.conf
--sa --usrp-tx-thread-config 1 -E --continuous-tx -d
--gNBs.[0].min_rxtxttime 6 --log_config.mac_log_level debug
|grep "coeff_ue" > coeficiente.log
```

El archivo generado con este gNB tiene líneas de esta forma, para el scheduler MaxTBS:

```
[NRMAC][UE5a2a][424,1]b2688,thr_ue3179,358154,tbs84,coeff_ue84,000000
[NRMAC][UE2a5b][424,2]b1153,thr_ue1118,531738,tbs22,coeff_ue22,000000
[NRMAC][UE5a2a][424,2]b6273,thr_ue3180,904785,tbs84,coeff_ue84,000000
[NRMAC][UE2a5b][424,3]b2817,thr_ue1119,380981,tbs22,coeff_ue22,000000
[NRMAC][UE5a2a][424,3]b0,thr_ue3179,314209,tbs84,coeff_ue84,000000
```

El problema que esta situación genera es que la salida estándar queda inutilizada y el uso del gNB con estas opciones queda “a ciegas”, sin la referencia de que está pasando que da el log en condiciones normales.

F.4. Otras Herramientas

T-trace

Una forma alternativa teóricamente útil de ver los logs sin perder la salida estándar es la herramienta T-Trace, que en teoría permite enviar los logs a una segunda salida, sin usar la salida estándar, y por lo tanto mandar a un archivo el log elegido, manteniendo la consola normal.

Esta herramienta se compila en forma independiente. La pudimos compilar, pero no logramos hacerla funcionar. Su uso es interesante, pero queda para el esfuerzo posterior. Un segundo uso de esta herramienta, es mandar a un puerto UDP, una copia de los mensajes de las capas MAC, RLC y PDCP. Wireshark está en condiciones de analizarlos desde ese puerto, según la documentación. No logramos que los mensajes se recibiesen en ese puerto. Por ahora todo el funcionamiento de esta herramienta no anda en nuestra instalación.

Apéndice F. Lecciones Aprendidas

Telnetserver

Otra herramienta disponible, que se compila aparte y sí funciona, es el telnetserver. Este servidor permite que mediante un telnet se pueda interactuar con un gNB o un UE que esté corriendo. Las interacciones existentes son limitadas. Es posible leer las opciones de configuración incluidas en el comando inicial, leer y modificar los niveles de log de las distintas categorías, y si se está corriendo un RF simulator con modelo de canal AWGN, se puede leer los valores configurados y setear el nivel de ruido blanco, variando su valor con el sistema funcionando.

Una vez compilado el telnetserver se puede correr en un gNB o UE agregando la opción `--telnetsrv`. Escucha por defecto en el puerto 9090. Se puede cambiar, por ejemplo para usar ese puerto en el gNB y agregar otro en el UE en la misma máquina (todo en uno) agregando por ejemplo `--telnetsrv.listenport 9095`.

Comando para usarlo: `telnet 127.0.0.1 9090`, desde la misma máquina. Desde la consola del telnet, “help” es la ayuda.

El comando “channelmod show current” muestra la configuración del modelo de canal en uso con RF simulator.

El comando `channelmod modify 1 noise_power_dB -5` varía la potencia de ruido en un gNB o UE con AWGN activado desde el inicio.

Además de esas pocas opciones, hay una librería con estructuras de datos y procedimientos, que permite agregar funciones de lectura o escritura de variables desde el telnetserver en el sw que está corriendo. Son modificaciones a hacer en el gNB o en el UE con el que se quiere interactuar.

F.5. ¡Ojo con las antenas!

Las antenas son livianas y tienen mucha movilidad, antes de comenzar a usarlas hay que repasar el ajuste de las conexiones coaxiales en antena y SDR porque se aflojan con el movimiento.

Un criterio práctico de ajuste, es apretar bien el conector con la mano. Apretarlo con una llave 8 no es recomendable, porque el bronce es blando y es fácil pasarse, deformando y rompiendo los conectores. Existen llaves torquimétricas para ajustar conectores SMA, pero no disponemos de ellas.

F.6. Conexiones coaxiales

En las pruebas con los SDR conectados por coaxial optamos por poner el atenuador de 30 dB en el transmisor siempre. Si se desconecta un cable, es suficiente el atenuador para evitar tener el transmisor con una onda reflejada total. **OJO NO CONECTAR NUNCA Tx con Rx de dos SDR SIN ATENUADORES de 30 dB mínimo.**

F.7. Conexión del PC con SDR X310

La conexión GB Ethernet es insuficiente para la maqueta. La pérdida de datos en la conexión con el SDR se indica en pantalla con muchas L que indican la pérdida de paquetes en esa conexión. LLLLLLLLLLLLLLLL. . .

Usando 10 GBE no debería pasar, pero a veces pasa. Revisar:

- MTU de la interfaz en 9000
- Setear todos los Cores de la PC en modo performance
- Ajuste del tamaño de los buffer de lectura y escritura a valores altos (ej, 62500000)
- Ajustar Ring Buffer de la Ethernet a 4096

Se pueden usarlos siguientes comandos:

```
for ((i=0;i<$(nproc);i++)); do sudo cpufreq-set -c $i -r -g
performance; done
sudo sysctl -w net.core.wmem_max=62500000
sudo sysctl -w net.core.rmem_max=62500000
sudo sysctl -w net.core.wmem_default=62500000
sudo sysctl -w net.core.rmem_default=62500000
sudo ethtool -G <Nombre-Interface-10G> tx 4096 rx 4096
```

A veces ocurre igual. Resetear el SDR puede ayudar. Algunas aplicaciones en la PC parecen incidir. Usando el ZOOM en la misma PC pasó muy seguido. A veces el uso de wifi USB en la PC también parece afectar.

En el caso del servidor que tiene 2 puertos 10 GBE, pasa que si uno se equivoca y conecta el SDR en el puerto con la IP de otra red que el SDR, puede funcionar igual. En el archivo de configuración se puede definir la conexión al SDR por tipo (type=X300) o por IP. Si está por tipo, lo busca, encuentra y funciona, pero a veces pierde paquetes y se desconecta. Las configuraciones que lo definen por IP, en el puerto correcto, son más estables en este sentido.

Arranque del gNB con SDR

Después de algunas veces de usado, hay que tirar el comando dos veces, porque la primera vez no arranca.

F.8. Conexión del PC con SDR B200

Es USB, y debe ser 3.0. Conexiones a puertos 2.0 no funcionan. A veces hay que desconectar y conectar el cable USB para que se conecte. El largo de los cables utilizados es un dolor de cabeza, son tan cortos que es difícil hallar una disposición física adecuada.

F.9. El problema del DNS

En las PC linux no logramos que tome DNS automáticamente de la interfaz virtual oaitun_ue1, por lo que en ese modo no se puede navegar por internet desde la PC UE por la red 5G, aunque tenga conectividad IP. Definir un DNS fijo externo funciona bien si la salida a internet del CORE es directa, por ejemplo usando un celular como modem usb. Pero si se sale por la red de facultad, si se fija un DNS externo, no funciona. Aparentemente no pasa por el proxy de la facultad. Usar el DNS de Facultad como DNS fijo, sí funciona bien.

Apéndice G

Glosario

G.1. Acrónimos

3GPP	3rd Generation Partnership Project
5GC	5G Core
AI	Artificial Intelligence
AMC	Adaptive Modulation and Coding
AMF	Authentication and Mobility Function
APN	Access Point Name
AR	Augmented Reality
BER	Bit Error Rate
BLER	Block Error Rate
BWP	Bandwidth Part
C-RNTI	Cell-Radio Network Temporary Identifier
CA	Carrier Aggregation
CC	Component Carrier
CN	Core Network
CORESET	Control Resource Set
COTS	Comercial off the shelves
CP	Control Plane
CQI	Channel Quality Indication
CRC	Cyclic Redundancy Code
CSI-RS	Channel State Information Reference Signal
D/A	Digital to Analogue
DC	Dual Connectivity
DL	Downlink
DMRS	Demodulation Reference Signal
DRB	Data Radio Bearer
DSS	Dynamic Spectrum Sharing
eMBB	Enhanced Mobile Broadband
eNB	Evolved NodeB
EPC	Evolved Packet Core
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform

FR1	NR Frequency Range 1 (debajo de 6 GHz)
FR2	NR Frequency Range 2 (sobre 6 GHz)
gNB	Next Generation NodeB
GTP	GPRS Tunnelling Protocol
GTP-UP	GPRS Tunnelling Protocol User Plane
GUAMI	Globally Unique AMF ID
IAB	Integrated Access and Backhauling
IFFT	Inverse Fast Fourier Transform
IIoT	Industrial IoT
IMSI	International Mobile Subscriber Identitiy
IoT	Internet of Things
ISI	Inter-symbol interference
ITS	Inteligent Transport Systems
ITU	International Telecommunication Union
KPI	Key Performance Indicator
LTE-M	LTE for Cat-M devices
LTE	Long Term Evolution
M2M	Machine to machine
MBB	Mobile Broadband
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
ML	Machine Learning
mMTC	Massive Machine type Communications
MU-MIMO	Multiple User MIMO
NAT	Network Address Translation
NB-IoT	Narrow Band IoT
NB	Narrow Band
NFV	Network Function Virtualization
NR	New Radio
NR-U	New Radio - Unlicenced band
NSA	Non Stand Alone
OFDM	Orthogonal frequency-division multiple
OFDMA	Orthogonal frequency-division multiple access
PAPR	Peak to Average Power Ratio
PBCH	Physical Bradcast Channel
PC	Personal Computer
PCC	Primary Component Carrier
PCI	Physical Cell Identifier
PDCCH	Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PF	Proportional Fair
PRB	Physical Resource Block
PSS	Primary Synchronization Signal
PUSCH	Physical Uplink Shared Channel
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RAT	Radio Access Technology
RB	Resource Block
RBG	Resource Block Group

Apéndice G. Glosario

RIC	RAN Intelligent Control
RR	Round Robin
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
SA	Stand Alone
SC-FDMA	Single Carrier frequency-division multiple access
SCC	Secondary Component Carrier
SCS	Sub carrier Spacing
SDN	Software Defined Networks
SINR	signal-to-interference-plus-noise ratio
SRB	Signalling Radio Bearer
SSB	Synchronization Signal Block
SSS	Secondary Synchronization Signal
SU-MIMO	Single User MIMO
SPGWU	Serving and PDN Gateway User plane function
TAC	Tracking Area Code
TB	Transport Block
TBS	Transport Block Size
TDD	Time Division Duplex
TRS	Tracking Reference Signal
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink
UP	User Plane
UPF	User Plane Function
URLLC	Ultra Reliable and Low Latency Communications
V2X	Vehicle to Anything
VPP	Virtual Power Plants
ZMQ	ZeroMQ library (Zero Message Queuing)

Referencias

- [1] 3GPP. 3gpp ts 38.201 v0.0.0. 3rd generation partnership project; technical specification group radio access network; nr; physical layer; general description (release 15). <https://www.itu.int/rec/R-REC-M.2083/es>, May 2017.
- [2] 3GPP. Nr; user equipment (ue) radio access capabilities (3gpp ts 38.306 version 15.3.0 release 15). https://www.3gpp.org/ftp/Specs/archive/38_series/38.306/38306-f30.zip, September 2018.
- [3] 3GPP. 3gpp ts 38.213 v15.7.0 nr; physical layer procedures for control (release 15). https://www.3gpp.org/ftp/Specs/archive/38_series/38.213/38213-f70.zip, September 2019.
- [4] 3GPP. 3gpp tr 21.916 version 16.0.1 release 16. <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3493>, 2021.
- [5] 3GPP. 3gpp ts 23.501 v15.13.0 system architecture for the 5g system (5gs); stage 2 (release 15). <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>, March 2022.
- [6] 3GPP. 3gpp ts 38.104 v18.3.0 base station (bs) radio transmission and reception (release 18). https://www.3gpp.org/ftp/Specs/archive/38_series/38.104/38104-i30.zip, September 2023.
- [7] 5G-Tools. 5g tools for rf wireless. <https://5g-tools.com/>.
- [8] OpenAirInterface Software Alliance. Proyectos de oai. <https://openairinterface.org/projects/>, December 2022.
- [9] AMARISOFT. Presentación de AMARI CALLBOX series. <https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/>, January 2023.
- [10] AvisThink. Myth-busting dpdk in 2020. revealed: the past, present, and future of the most popular data plane development kit in the world. <https://nextgeninfra.io/dpdk-myth-busting-2020/>, 2020.

Referencias

- [11] Pablo Belzarena and Federico La Rocca. Comunicaciones inalámbricas. <https://iie.fing.edu.uy/investigacion/grupos/artes/es/investigacion/libro-comunicaciones-inalambricas/>, October 2017.
- [12] Jessy Cavazos. The aspects of 6g that will matter to wireless design engineers. <https://www.ednasia.com/the-aspects-of-6g-that-will-matter-to-wireless-design-engineers/>.
- [13] Erik Dahlman, Stefan Parkvall, and Johan Sköld. *5G NR: the Next Generation Wireless Access Technology*. Academic Press, 2018.
- [14] Raphael Defosseux. Deploy sa5g basic deployment. https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_BASIC_DEPLOYMENT.md, June 2023.
- [15] Robert Schmidt Florian Kaltenberger, Tien Thinh Nguyen. Roadmaps for ran, core network, o&m, and ric activities at openairinterface, eurecom & oai. <https://openairinterface.org/wp-content/uploads/2022/11/6a-RAN-roadmap.pdf>, November 2022.
- [16] Robert Schmidt Florian Kaltenberger, Tien Thinh Nguyen. Miembros de la openairinterface software alliance. <https://openairinterface.org/osa-members/>, January 2023.
- [17] free5GC. The free5gc is an open-source project for 5th generation (5g) mobile core networks. the ultimate goal of this project is to implement the 5g core network (5gc) defined in 3gpp release 15 (r15) and beyond. <https://www.free5gc.org/>, January 2023.
- [18] Francesco Gringoli, Paul Patras, Carlos Donato, Pablo Serrano, and Yan Gruenberger. Performance assessment of open software platforms for 5g prototyping. *IEEE Wireless Communications*, 25(5):10–15, 2018.
- [19] GSMA. Operator requirements for 5g core connectivity options. <https://www.gsma.com/futurenetworks/wiki/operator-requirements-for-5g-core-connectivity-options/>, May 2019.
- [20] GSMA. The 6 ghz int ecosystem. demand drives scale. <https://www.gsma.com/spectrum/wp-content/uploads/2022/07/6-GHz-in-the-5G-Era.pdf>, August 2022.
- [21] Harri Holma, Antti Toskala, Takehiro Nakamura, and Tommi Uitto. *5G Technology*. John Wiley & Sons, Ltd, 2020.
- [22] Sadiq Iqbal and Jehad M. Hamamreh. A Comprehensive Tutorial on How to Practically Build and Deploy 5g Networks Using Open-Source Software and General-Purpose, Off-the-Shelf Hardware. *RS Open Journal on Innovative Communication Technologies*, 2(6), Dec 16 2021. <https://rs-ojct.pubpub.org/pub/lqhigiva>.

- [23] ITU-R. Recomendación uit-r m.2083-0. concepción de las imt – marco y objetivos generales del futuro desarrollo de las imt para 2020 y en adelante. <https://www.itu.int/rec/R-REC-M.2083/es>, September 2015.
- [24] Florian Kaltenberger, Aloizio P. Silva, Abhimanyu Gosain, Luhan Wang, and Tien-Thinh Nguyen. Openairinterface: Democratizing innovation in the 5g era. *Computer Networks, Volume 176, 20 July 2020*, 2020. © Elsevier. Personal use of this material is permitted. The definitive version of this paper was published in Computer Networks, Volume 176, 20 July 2020 and is available at : <https://doi.org/10.1016/j.comnet.2020.107284>.
- [25] Francasco Mani. Openairinterface. sw_archi.md. https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/SW_archi.md?ref_type=heads, June 2023.
- [26] FlexRAN MOSAIC5G. Openlte is an open source implementation of the 3gpp lte specifications. <https://mosaic5g.io/flexran/>, January 2023.
- [27] NVIDIA. Presentación de NVIDIA aerial, un framework de desarrollo. <https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/>, January 2023.
- [28] Open5GS. Open5gs is a c-language open source implementation for 5g core and epc, i.e. the core network of lte/nr network (release-16). <https://open5gs.org/open5gs/>, January 2023.
- [29] OpenLTE.sourceforge. Openlte is an open source implementation of the 3gpp lte specifications. <https://openlte.sourceforge.net/>, January 2023.
- [30] Janne Peisa, Stefan Persson, Patrik Parkvall, Erik Dahlman, Asbjorn Ggrolven, Chtistian Hoymann, and Dirk Gerstenber. 5g evolution: 3gpp releases 16 and 17 overview. Technical report, Ericsson Technology Review, October 2017.
- [31] Gabriela Pereyra. Scheduling in 5g networks: Developing a 5g cell capacity simulator, 2021. Tesis de Maestría. UDELAR, Facultad de Ingeniería. Montevideo, Uruguay.
- [32] Pablo Picazo Martínez. *Despliegue de redes 5G Non-Standaone basadas en OpenAirInterface*. PhD thesis, Universitat Politècnica de València, 2021.
- [33] Ettus Research. Ettus research™, an ni brand since 2010, is the world’s leading supplier of software-defined radio platforms, including the universal software radio peripheral (usrp™) family of products. <https://www.ettus.com/>, January 2023.
- [34] Sharetechnote. https://www.sharetechnote.com/html/5G/5G_HARQ.html.
- [35] ShareTechnote. 5g/nr - frame structure. https://sharetechnote.com/html/5G/5G_FrameStructure.html.

Referencias

- [36] Maraj Uddin Ahmed Siddiqui, Faizan Qamar, Syed Hussain Ali Kazmi, Rosilah Hassan, Asad Arfeen, and Quang Ngoc Nguyen. A study on multi-antenna and pertinent technologies with ai/ml approaches for b5g/6g networks. *electronics MDPI*, 2023.
- [37] sqimway. Radio calculator. https://www.sqimway.com/rb_calc.php.
- [38] srsRAN. 5g nsa end-to-end. https://docs.srsran.com/en/latest/app_notes/source/5g_nsa_zmq/source/index.html, December 2022.
- [39] srsRAN. Presentación srsRAN. <https://docs.srsran.com/en/latest/>, December 2022.
- [40] srsRAN. srsran with zmq virtual radios. https://docs.srsran.com/en/latest/app_notes/source/zeromq/source/index.html, December 2022.
- [41] Techplayon. <https://www.techplayon.com/5g-nr-transport-block-size-tbs-calculation/>.
- [42] Laurent Thomas. Open cells project. <https://open-cells.com/>, February 2023.
- [43] UIT. Recomendación uit-r m.2150. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2150-0-202102-S!!PDF-S.pdf, February 2021. 1.1.2.4.1 Forma de onda, numeración y estructura de trama.
- [44] Pratheek S. Upadhyaya, Aly S. Abdalla, Vuk Marojevic, Jeffrey H. Reed, and Vijay K. Shah. Prototyping next-generation o-ran research testbeds with sdrs, 2022.

Índice de tablas

3.1. Tabla comparativa de características técnicas entre srsRAN y OAI a la fecha de la selección.	44
4.1. Costo del hardware utilizado en este proyecto.	58
4.2. Horas de trabajo empleadas en la realización del proyecto.	59
4.3. Listado del software utilizado en la implementación definitiva de la maqueta.	59
6.1. Tabla de performances medidas con dos UE usando RFsim, en una celda de 40 MHz de ancho de banda (106 PRB). Se comparan los resultados obtenidos con scheduler Proportional Fair y MaxTBS.	94
6.2. Pruebas de performance con scheduler Proportional Fair y MaxTBS utilizando SDR, en una celda de 10 MHz (24 PRB).	95
C.1. Tabla Primer Set de Pruebas, en aire y cable coaxial. Se agregan valores teóricos para cada número de PRB.	141
C.2. Tabla Primer Set de Pruebas, RFsim.	143
C.3. Tabla Segundo Set de Pruebas, 2 UE B200.	143
C.4. Tabla Segundo Set de Pruebas, RFsim Sin Ruido.	144
C.5. Tabla Segundo Set de Pruebas, MaxTBS con Ruido.	144
C.6. Tabla de throughput teórico según la fórmula presentada en C.1.	147
E.1. Tabla de bandas de frecuencias FR1.	157
E.2. Tabla de bandas de frecuencias FR1, continuación.	158
E.3. Tabla de bandas de frecuencias FR2.	158

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

1.1. Diagrama general de la maqueta propuesta, que incluye Core Network y una radio base funcionando en una computadora con un SDR como interfaz de radio y un terminal de usuario en otra computadora, también con SDR.	4
2.1. Ejemplos de casos de aplicación de 5G, según la clasificación propuesta por la ITU en la Recomendación M.2083-0 [23].	11
2.2. Distintas formas de conexión de los planos de usuario y de control en despliegues Non Stand Alone con las redes 4G y 5G interdependientes y Stand Alone con las redes 4G y 5G completamente separadas. En este proyecto se implementa la opción 2, 5G SA con la tecnología NR completamente en la RAN y el core 5G. En las demás opciones se utiliza LTE ya sea en el plano de control o de usuario o ambos en combinación con NR.	13
2.3. Duplexado en tiempo y frecuencia [36].	19
2.4. Construcción de OFDM a partir de su espectro. (Figura elaborada a partir de Keysight, Flexible OFDM Signal Generation, Analysis and Troubleshooting - Application Note.)	21
2.5. Esquema de bloques de OFDM. Se indican en azul las funciones hechas por los SDR [11].	22
2.6. Estructura de trama, subtrama y slot SCS 15khz. [35]	23
2.7. Estructura de trama, subtrama y slot SCS 30khz [35]	23
2.8. Duración de símbolo OFDM según la numerología.	24
2.9. Configuración de símbolos asignados a UL y DL dentro de un slot híbrido [3]. (figura de sharetechnote.com	25
2.10. Diagrama de estados del Radio Resource Control. [13]	26
2.11. Diagrama de fragmentación del Bloque de Transporte en capa RLC. [41]	27
2.12. Diagrama simplificado para obtener el tamaño del bloque de transporte. Proporciona listas de parámetros que afectan el tamaño del TB.	28
2.13. Diagrama de funcionamiento de HARQ. [34]	31
2.14. Los paradigmas de 6G, implica la incorporaciones de frecuencias de THz y de AI. [12]	33

Índice de figuras

3.1.	<i>RAN Roadmap 2022/2023</i> presentado en el <i>Fall 2022 OpenAirInterface North American Workshop</i> . Presenta las perspectivas de desarrollo en el momento en que se tomó la decisión.	41
3.2.	<i>RAN Roadmap 2022/2023</i> , continuación.	41
3.3.	Conexión de prueba sin radios entre un UE y un gNB en una instalación preliminar de la plataforma OAI.	43
4.1.	Diagrama de bloques del Core Network OAI instalado [14].	46
4.2.	Diagrama general de la maqueta implementada, con Core Network y gNB funcionando en una computadora con un SDR como interfaz de radio y un terminal de usuario en otra computadora, también con SDR.	47
4.3.	Frente y fondo del SDR X310, indicando las interfaces de conexión utilizadas.	49
4.4.	Las tarjetas SIM utilizadas en el proyecto, junto con la grabadora SIM USB.	51
4.5.	Celular samsung A33 conectado y registrado, en la parte superior se muestra el MCC, MNC junto al nombre de la maqueta y el logo del 5G.	52
4.6.	Espectro de un gNB con 40 Mhz de ancho de banda en la banda n41.	53
4.7.	Log del Authentication and Mobility Function, donde se ve el estado de los gNB de la red y los UE con su estado de conexión.	55
4.8.	Scope de un UE mostrando las señales recibidas en Down Link.	56
4.9.	Maqueta en operación con dos SDR X310.	57
5.1.	Logs del core AMF, donde se muestran el gNB conectado y el UE registrado.	62
5.2.	Captura de NGAP en conexión de un UE.	62
5.3.	Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y UE conectados por cable 10 Gb Ethernet, utilizando RFSim.	65
5.4.	Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y UE en SDR con antenas.	65
5.5.	Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y un UE en SDR conectados por coaxial.	66
5.6.	Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y 2 UE en SDR, y servidor Iperf en un PC externo a la red	66
5.7.	Esquema de dispositivos y conexiones para pruebas de rendimiento con un gNB y 2 UE en SDR, servidor Iperf en uno de los UE o sea interno a la red.	67
5.8.	Throughput según medio y PRB, se comparan datos obtenidos de la transmisión por aire y coaxial, comparados con los máximos teóricos.	69
5.9.	Throughput en RFSim Según PRB donde se muestran los datos obtenidos, en los que se observan rendimientos muy por encima de lo teórico para los anchos de banda más bajos.	71

5.10. Comportamiento de la CPU y del tráfico ethernet durante la transmisión en RFSim.	72
5.11. Throughput aire 2 UE, según ubicación del servidor iperf, interno (color verde) o externo a la red (color azul).	73
5.12. Throughput RFSim 2 UE, transmitiendo solos (color azul) y en forma simultánea (color verde).	74
5.13. Speed test de internet, de móvil samsung A33 conectado al Core a través de SDR X310. El core esta conectado al internet libre de la Facultad de Ingeniería	75
5.14. Desempeño de la cpu en el servidor del core cuando la plataforma esta corriendo con un gNB y 2 UE conectados transmitiendo datos.	76
6.1. Multiplexado de datos de capa MAC en Down Link, para el armado del Transport Block [13].	78
6.2. Estructura de archivos en git RAN OAI.	79
6.3. Esquema general de hilos de tareas en el sistema OAI-RAN.	81
6.4. Diagrama de flujo de la definición de prioridades de los UE conectados a un gNB en el preproceso del scheduler.	83
6.5. Diagrama de flujo de la definición de prioridades de los UE conectados a un gNB en el preproceso del scheduler modificado.	85
6.6. Función de comparación para el ordenado de la lista de UE a agendar.	85
6.7. Esquema de maqueta para pruebas de scheduler con RFSim.	88
6.8. Scope de los dos UE, uno con ruido y el otro no.	89
6.9. Esquema de maqueta para pruebas de scheduler con tres SDR.	90
6.10. Asignación de Bytes en dos UE con Proportional Fair.	91
6.11. Asignación de Bytes en dos UE con MaxTBS.	91
6.12. Asignación aleatoria de Bytes a dos UE con igual MCS en MaxTBS.	92
6.13. Alternancia de la asignación en Proportional Fair en régimen. Se observa que en la misma trama se asigna un slot a un UE identificado por su RNTI, y el siguiente al otro.	92
6.14. Asignación en Max TBS con distinto MCS. Se puede ver que casi todos los slots son asignados al UE con MCS mayor.	93
6.15. Asignación en Max TBS con igual MCS. La adjudicación de slots es pseudoaleatoria.	93
6.16. Asignación a dos UE durante pruebas de ping con 1000 Bytes. Se puede ver que casi todos los slots quedan sin asignarse. Es indiferente el scheduler utilizado.	94
A.1. USRP X310.	102
A.2. UBX 160.	103
A.3. LP0965 Antenna.	104
A.4. Tarjeta 10 Gbps.	104
A.5. Cable SFP.	105
A.6. Gráficos UE estatus	113
A.7. Gráficos gNB estatus	114
A.8. Logs Core AMF	114

Índice de figuras

A.9. Espectro de radioemisora a 95.5 MHz y entorno usando la aplicación dft del driver UHD.	131
A.10. Señal de 1 GHz transmitida por el SDR bajo prueba y verificada por un segundo SDR, con la herramienta del driver rx_ascii_art_dft.	132
B.1. NR configuración.	133
B.2. gNB conectado al core.	136
B.3. ue conectado al core.	136
B.4. gNB traza.	137
B.5. UE traza.	137
B.6. UE, traza iperf.	138
B.7. gNB traza iperf.	138
C.1. Throughput según medio y PRB del primer set de pruebas, se contrasta con datos máximos teóricos	142
C.2. Throughput en RFsim Según PRB.	145
C.3. Throughput en Aire Según ubicación de servidor iperf.	145
C.4. Throughput en RFsim Según tipo de Algoritmos de scheduler.	146
C.5. Throughput en RFsim, con MaxTBS y ruido.	146

Esta es la última página.
Compilado el sábado 10 febrero, 2024.
<http://iie.fing.edu.uy/>