ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Máster

GESTIÓN CONJUNTA DE RECURSOS Y DIVISIÓN FUNCIONAL EN REDES C-RAN

(Joint resource management and functional split in C-RAN)

Para acceder al Título de

Máster Universitario en Ingeniería de Telecomunicación

Autor: Sofía Pérez Rivas

Septiembre- 2023



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE MÁSTER

Realizado por: Sofía Pérez Rivas

Director del TFG: Luis Francisco Díez Fernández, Ramón Agüero Calvo **Título:** "Gestión conjunta de recursos y división funcional en redes C-RAN" **Title:** "Joint resource management and functional split in C-RAN"

Presentado a examen el día: 26 de Septiembre de 2023

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del tribunal:

Presidente (Apellidos, Nombre): Muñoz Gutiérrez, Luis Secretario (Apellidos, Nombre): Menéndez de Llano Rozas, Rafael Vocal (Apellidos, Nombre): García Arranz, Marta

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFM (solo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Máster N^o (a asignar por Secretaría)

Agradecimientos

Me gustaría agradecer a mis directores de TFM, Ramón Agüero Calvo y Luis Francisco Díez Fernández, por todo el apoyo y ayuda recibidos para la realización de este proyecto. La confianza que han depositado en mí y todos los conocimientos que me han enseñado han hecho posible este trabajo.

A mi familia, especialmente a mis padres y a mi abuela Susi, por todo el apoyo y ánimo recibido durante esta etapa, por creer en mí y por respaldarme en mis decisiones. Asimismo, a mi tío Fernando, por adentrarme en el mundo de las telecomunicaciones y animarme a escoger esta carrera.

A mis amigos, por estar ahí y acompañarme a lo largo de estos años.

Gracias a todos.

Resumen

En este trabajo se plantea un problema de optimización que aborda dos aspectos fundamentales: la gestión de los recursos radio y la selección de la división funcional en arquitecturas C-RAN. Se desarrollan e implementan algoritmos inteligentes que proporcionan soluciones óptimas a dicho problema. Se plantean dos alternativas centradas en la minimización de la asignación de recursos de las estaciones base, basadas en dos aproximaciones diferentes, y se diseña un tercer algoritmo que resuelve además la selección de la división funcional de forma dinámica. Se consideran diversos escenarios de aplicación para la validación de los tres algoritmos, diferenciándose en el número de estaciones base desplegadas y en las áreas existentes. Se crea un entorno de simulación que permite la evaluación de los algoritmos desarrollados, así como la automatización de su ejecución para facilitar la realización de experimentos. Este entorno posibilita la generación de resultados y su visualización de forma gráfica para su posterior análisis. A partir de esta herramienta, se realizan un amplio número de experimentos, con la finalidad de analizar el rendimiento de los algoritmos propuestos. **Palabras clave:** 5G; C-RAN; functional split; network slicing

Abstract

In this project, we contemplate an optimization problem that covers two fundamental aspects: the management of the radio resources and the selection of the functional split in C-RAN architectures. Intelligent algorithms that provide optimal solutions to such problem are designed and implemented. Two algorithms, which aim at the reduction of the resources allocated by the base stations, based on two different approximations, are proposed, and a third algorithm, to also tackle the dynamic selection of the functional split, is then discussed. Diverse application scenarios are considered for the validation of the implemented algorithms, distinguishing from one another in the number of deployed base stations and in the underlying areas. A simulation environment is created with the goal of allowing the evaluation of the developed algorithms, as well as the automation of its execution. This environment enables the generation of results and its graphical visualization for its corresponding analysis. By exploiting this tool, it is possible to conduct a large number of experiments, with the purpose of analyzing the performance of the proposed algorithms.

Keywords: 5G; C-RAN; functional split; network slicing

Índice general

Ín	Índice de figuras 8		
Ín	dice d	e tablas	10
Ín	dice d	e acrónimos	11
1	Intro	oducción	15
	1.1.	Motivación	15
	1.2.	Objetivos	16
	1.3.	Estructura de la memoria	16
2	Esta	do del arte	17
	2.1.	Evolución de las redes móviles	17
		2.1.1. 1 ^a Generación (1G)	17
		2.1.2. 2 ^a Generación (2G)	17
		2.1.3. 3ª Generación (3G)	18
		2.1.4. 4 ^a Generación (4G)	18
		2.1.5. 5 ^a Generación (5G)	19
	2.2.	Arquitectura C-RAN	22
	2.3.	Functional split	25
		2.3.1. Descripción de los <i>functional splits</i> posibles entre la CU y la DU	26
	2.4.	Network slicing	30
3	Plan	teamiento del problema	33
	3.1.	Requisitos del enlace fronthaul para cada split	33
	3.2.	Descripción del problema	35
		3.2.1. Modelo del sistema	36
		3.2.2. Formulación del problema	38
	3.3.	Escenarios de aplicación	41
4	Desa	urrollo e implementación	43
	4.1.	Parámetros de configuración	43

4.2. Algoritmos implementados			tmos implementados	44			
		4.2.1.	GLPK	44			
		4.2.2.	Primer algoritmo	45			
		4.2.3.	Segundo algoritmo	45			
		4.2.4.	Tercer algoritmo	46			
	4.3.	Entorn	o de simulación	48			
		4.3.1.	Clases implementadas	48			
		4.3.2.	Automatización de la plataforma y herramientas utilizadas	52			
5	Resu	ultados		55			
	5.1.	Valida	ción del algoritmo	55			
		5.1.1.	Colas virtuales de las zonas del primer escenario	56			
		5.1.2.	Throughput y cola virtual de la zona 6 del primer escenario	58			
	5.2. Análisis de los tres escenarios de aplicación						
		5.2.1.	Recursos y throughput medios para los tres escenarios	62			
		5.2.2.	Normalización de los recursos medios asignados en el tercer escenario	64			
	5.3.	Estudi	o de la implementación de niveles de split dinámicos	65			
		5.3.1.	Recursos medios en función de la complejidad computacional	66			
		5.3.2.	Probabilidad de selección de cada nivel de <i>split</i>	67			
6	Con	clusion	es	68			
	6.1.	Líneas	futuras	69			
Bi	Bibliografía 7			70			

Índice de figuras

2.1.	Casos de uso 5G	20
2.2.	Arquitectura NSA	21
2.3.	Arquitectura SA	22
2.4.	Arquitectura D-RAN	22
2.5.	Arquitectura C-RAN	23
2.6.	Arquitectura C-RAN con la implementación del functional split	24
2.7.	Pila de protocolos de una BS LTE	25
2.8.	Functional splits posibles entre la CU y la DU definidos por el 3GPP	26
2.9.	Casos de uso de los <i>slices</i> obtenidos gracias al <i>Network slicing</i>	31
2.10.	Arquitectura del Network slicing	32
3.1.	Escenario de referencia	36
3.2.	Primer escenario de aplicación	41
3.3.	Segundo escenario de aplicación	41
3.4.	Tercer escenario de aplicación	42
4.1.	Diagrama de clases de los algoritmos implementados	44
4.2.	Diagrama de flujo de los dos primeros algoritmos	46
4.3.	Diagrama de flujo del tercer algoritmo	48
4.4.	Diagrama de flujo de la clase <i>Maestra</i>	51
4.5.	Esquema de la plataforma implementada	53
5.1.	Evolución de las colas virtuales en función del <i>slot</i> para el primer algoritmo	56
5.2.	Evolución de las colas virtuales en función del <i>slot</i> para el segundo algoritmo	57
5.3.	Throughput y cola virtual de la zona 6 en función del throughput mínimo y de V para el	
	primer algoritmo	58
5.4.	Throughput y cola virtual de la zona 6 en función del throughput mínimo y de V para el	
	segundo algoritmo	60
5.5.	Recursos medios asignados a cada <i>slice</i> en función de los niveles de centralización y de los	
	escenarios	62
5.6.	Distribución del throughput medio en función de los niveles de centralización y de los	
	escenarios considerados	64

5.7. Recursos medios normalizados asignados a cada slice por cada Base Station (BS) física de		
	tercer escenario para los cuatro niveles de centralización	65
5.8.	Distribución de los recursos medios en función de la complejidad computacional para la	
	eficiencia espectral de 4G y de 5G	66
5.9.	Probabilidad de selección de cada nivel de split en función del área y de la complejidad	
	computacional de la CU	67

Índice de tablas

3.1.	Requisitos del <i>split</i> intra MAC en términos de latencia	34
3.2.	Ratios y complejidad computacional de cada <i>functional split</i>	35
3.3.	Símbolos y definiciones	37
4.1.	Factor de cancelación de la interferencia para cada <i>functional split</i>	47
5.1.	Parámetros de configuración	55

Índice de acrónimos

1G	1ª Generación.	
2G	2ª Generación.	
3G	3ª Generación.	
3GPP	Third Generation Partnership Project.	
4G	4ª Generación.	
5G	5ª Generación.	
AMPS	Advanced Mobile Phone System.	
ARQ	Automatic Repeat Request.	
BBU	Baseband Unit.	
BER	Bit Error Rate.	
BS	Base Station.	
CDMA-2000	Code Division Multiple Access-2000.	
CoMP	Coordinated MultiPoint.	
CPRI	Common Public Radio Interface.	
C-RAN	Cloud Radio Access Network.	
CU	Centralized Unit.	
~ *		
DL	downlink.	
D-RAN	Distributed Radio Access Network.	
DU	Distributed Unit.	
FDGE	Enhanced Data Rates for GSM Evolution	
oMBB	Enhanced Mahila Broadhand	
oNP	eMadeP	
	Evolued Dealest Core	
EPC	Evolved Packet Core.	
E-UTRA	Evolved Universal Terrestrial Radio Access.	

FDMA	Frequency Division Multiple Access.	
FR1 Frequency Range 1.		
FR2	Frequency Range 2.	
GLPK	GNU Linear Programming Kit.	
gNB	gNodeB.	
GOPS	Giga Operaciones por Segundo.	
GPRS	General Packet Radio Service.	
GSM	Global System for Mobile communications.	
HARQ	Hybrid Automatic Repeat Request.	
HD	High Definition.	
HSDPA	High Speed Downlink Packet Access.	
HSUPA	High Speed Uplink Packet Access.	
IMT-2000	International Mobile Telecommunication-2000.	
IoT	Internet of Things.	
IP	Internet Protocol.	
ITU	International Telecommunication Union.	
ITU-R	Radiocommunication Sector of the International	
	Telecommunication Union.	
LTE	Long Term Evolution.	
LTE-A	Long Term Evolution Advanced.	
M2M	machine to machine.	
MAC	Medium Access Control.	
MIMO	Multiple Input Multiple Output.	
MMS	Multimedia Messaging Service.	
mMTC	Massive Machine Type Communications.	
mmWave	ondas milimétricas.	
NaaS	Network as a Service.	
NFV	Network Function Virtualization.	
NG-RAN	Next Generation Radio Access Network.	
NMT	Nordic Mobile Telephone.	
NR	New Radio.	
NSA	Non-Stand Alone.	
OFDMA	Orthogonal Frequency Division Multiple Access.	
PDCP	Packet Data Control Protocol.	

PDU	Protocol Data Units.
PHY	Physical.
PRACH	Physical Random Access Channel.
PRB	Physical Resource Block.
QoS	Quality of Service.
RAN	Radio Access Network.
KB	Resource Block.
KF DL G	radiofrecuencia.
RLC	Radio Link Control.
RRC	Radio Resource Control.
RRH	Remote Radio Head.
RRU	Remote Radio Unit.
RTT	Round Trip Time.
SA	Stand Alone
SDN	Software Defined Network
SINR	Signal to Interference plus Noise Ratio
SLA	Service Level Agreement
SMS	Short Message Service
SNR	Signal to Noise Ratio
SINK	Signal to Noise Ratio.
TACS	Total Access Communication System.
TDMA	Time Division Multiple Access.
TD-SCDMA	Time Division Synchronous Code Division Multi-
	ple Access.
IIF	User Equipment
	unlink
UMTS	Universal Mobile Telecommunication System
	Ultra Paliable Low Latency Communications
UKLLC	Unita Reliable Low Latency Communications.
VNF	Virtual Network Functions.
VNI	Virtual Network Instances.
VoLTE	Voice over Long Term Evolution.
ΨΑΡ	Wireless Application Protocol
	Wideband Code Division Multiple Access
	Worldwide Interenerability for Microwaya Access
WINIAA	wondwhile interoperating for Microwave Access.

Capítulo I

Introducción

1.1. Motivación

La implantación de la tecnología 5G habilita nuevos escenarios de comunicaciones y servicios heterogéneos, que presentan nuevas necesidades en términos de velocidad de transmisión, ancho de banda, latencia y seguridad, entre otros [1]. Algunos de estos servicios son la realidad virtual, la realidad aumentada o el vídeo HD en tiempo real, precisando de una velocidad elevada de datos para su provisión. Así mismo, 5G posibilita la ejecución de aplicaciones críticas que requieren una baja latencia y una conectividad altamente confiable con bajas tasas de error como, por ejemplo, el control remoto de equipos sensibles. Por otro lado, esta tecnología permite el despliegue del internet de las cosas, donde se requiere la conexión de una elevada cantidad de dispositivos con alta conectividad entre sí y donde se precisa de una baja latencia en la transmisión de información [2].

Las arquitecturas de las generaciones móviles precursoras a 5G no son capaces de satisfacer las demandas establecidas por los nuevos servicios, por lo que se recurre a las arquitecturas 5G para dar solución a este problema. Concretamente, estas últimas arquitecturas implementan las tecnologías *Software Defined Network/Network Function Virtualization (SDN/NFV)* para habilitar la virtualización de las funciones de las estaciones base, y la arquitectura *Cloud Radio Access Network (C-RAN)*, en la que se computan las funciones virtualizadas en la nube, de forma centralizada.

La arquitectura C-RAN presenta dos elementos fundamentales: *Baseband Unit (BBU)* y Remote Radio Unit (RRU). La unidad BBU se encarga de realizar las funciones de procesado en banda base y la unidad RRU se centra en las funciones radio y parte de las funcionalidades de la capa física, estando situada junto a las antenas [3]. Posteriormente, la BBU se divide en dos elementos: *Centralized Unit (CU)*, que realiza el procesado de las funciones de forma centralizada, y *Distributed Unit (DU)*, que lleva a cabo las funciones más cercanas a las estaciones base [4]. La decisión de qué funcionalidades se realizan en la CU y cuales se procesan en la DU introduce el concepto del *functional split*, que supone una división vertical de las funciones [5].

La integración de los servicios con diferentes requisitos de rendimiento sobre una misma infraestructura física en las redes 5G conlleva la introducción del concepto de *Network Slicing*. Este término se refiere a la separación de una red física en múltiples redes virtuales, siendo cada una de ellas diseñada y optimizada para un servicio concreto. La incorporación de este concepto a las redes 5G posibilita la provisión de

nuevos servicios de forma óptima.

1.2. Objetivos

El desarrollo de este trabajo se centra en la gestión de los recursos radio y en la división funcional en arquitecturas C-RAN. Para abordar estos dos aspectos, se presentan a continuación los objetivos principales sobre los que se articular este trabajo:

- El desarrollo, la implementación y la evaluación de algoritmos inteligentes que permitan minimizar el número de recursos radio asignados por las estaciones base, y realizar una selección del *functional split* de forma dinámica.
- La descripción y la implementación de diversos escenarios de aplicación con diferentes parámetros de configuración, que posibiliten la evaluación de los algoritmos desarrollados.
- La generación de un entorno de simulación que facilite la caracterización del funcionamiento de dichos algoritmos, así como la automatización de la ejecución de los mismos y de la generación de resultados derivados de su aplicación.

1.3. Estructura de la memoria

Este trabajo se ha dividido en seis capítulos, detallando el primero de ellos la motivación y los objetivos del proyecto. El resto del documento se estructura de la siguiente manera:

- El Capítulo 2 proporciona el marco teórico en el que se desarrolla el trabajo. Realiza una descripción de la evolución de las redes móviles, desde la primera hasta la quinta generación, detalla las características de la arquitectura C-RAN, e introduce los conceptos de *functional split* y *network slicing*.
- El Capítulo 3 lleva a cabo el planteamiento del problema a resolver, así como la descripción de los escenarios de aplicación utilizados para ello. Especifica los requisitos teóricos del enlace *fronthaul* para cada *functional split* en términos de *throughput* y latencia, y presenta el término de complejidad computacional.
- En el Capítulo 4 se explica el desarrollo e implementación de los algoritmos creados para la resolución del problema planteado, así como la generación y la aplicación del entorno de simulación desarrollado para la evaluación de los mismos. Se indican los parámetros de configuración necesarios para el funcionamiento adecuado de este entorno y de los algoritmos.
- El Capítulo 5 incluye la presentación y la explicación de los resultados obtenidos tras ejecutar los algoritmos implementados para los escenarios de aplicación considerados. Estos resultados permiten realizar la validación de las soluciones propuestas, y estudiar el impacto de la aplicación de *functional splits* dinámicos.
- En el Capítulo 6 se recogen las conclusiones obtenidas de la realización de este trabajo, y se incluyen varias líneas de investigación futuras que le dan continuidad.

Capítulo 2

Estado del arte

En este capítulo se describe, de forma breve, la evolución de las redes móviles, desde la aparición de la primera generación hasta la llegada de la quinta generación. Además, se detalla en profundidad la arquitectura C-RAN, junto con los conceptos de *functional split* y *network slicing*.

2.1. Evolución de las redes móviles

Este apartado recoge las principales características de las redes móviles, abarcando desde la primera generación hasta la quinta, para contextualizar el trabajo.

2.1.1. 1^ª Generación (1G)

A principios de la década de 1980 surge la l^a *Generación* (*1G*) de tecnología celular móvil con el objetivo de proporcionar llamadas de voz. 1G se basa en un sistema analógico de comunicación que admite velocidades de 2.4 Kbps y que dispone de una red de conmutación de circuitos [2, 6].

El sistema se utiliza solo para servicios de voz por medio del uso de la técnica *Frequency Division Multiple Access (FDMA)*. Las frecuencias operativas se sitúan en los 800 MHz y 900 MHz, presentando una capacidad de canal limitada a 30 KHz [5]. En 1G se implementan los estándares tecnológicos *Advanced Mobile Phone System (AMPS)* en los Estados Unidos, *Nordic Mobile Telephone (NMT)* en las naciones nórdicas, y *Total Access Communication System (TACS)* en el Reino Unido [2, 6].

1G dispone de una infraestructura simple y necesita pocos elementos de red para su desarrollo [5]. Presenta diversos inconvenientes, como la baja velocidad de datos y la falta de seguridad, debido a que la llamada de voz se guarda y se reproduce en torres de radio, lo que incrementa el riesgo de ser escuchada por parte de terceros no acreditados [2]. Las baterías no rinden con eficiencia, la recepción de las llamadas no es la esperada y se producen interferencias de ruido de fondo [5].

2.1.2. 2ª Generación (2G)

La 2^a Generación (2G) de telefonía móvil aparece a finales de 1990. El primer sistema estándar de 2G que incorpora tecnología digital es *Global System for Mobile communications (GSM)*, que proporciona servicios de voz, *Short Message Service (SMS)*, *Multimedia Messaging Service (MMS)* y correo electrónico. GSM se basa en la tecnología de acceso *Time Division Multiple Access (TDMA)* y en conmutación de

circuitos. GSM opera a frecuencias de 900 MHz y 1800 MHz, ofrece anchos de banda que van desde los 25 KHz hasta los 200 KHz y brinda una tasa de datos máxima de 64 Kbps [2, 5].

GSM mejora las comunicaciones de voz a través de la utilización de codificadores de voz o *vocoders*, aumenta la capacidad del sistema y refuerza la seguridad en la transmisión de datos mediante el uso de cifrado digital [5, 6].

Con el fin de aumentar las tasas de datos de los usuarios por encima de las de GSM, y proporcionar la capacidad de conmutación de paquetes, 2G evoluciona hacia otras dos extensiones: la primera, llamada *General Packet Radio Service (GPRS)* o 2.5G, mejora la velocidad de datos hasta los 150 Kbps y se utiliza para que los usuarios tengan acceso a servicios *Wireless Application Protocol (WAP)*/Internet, y la segunda, nombrada como *Enhanced Data Rates for GSM Evolution (EDGE)* o 2.75 G, proporciona una tasa de datos máxima de 384 Kbps [2, 5, 6].

2.1.3. 3ª Generación (3G)

A finales del año 2000 aparece la 3^a Generación (3G) de sistemas de comunicación móvil para proveer servicios mejorados más rápidos y confiables. 3G surge a partir del trabajo de la International Telecommunication Union (ITU) mediante el estándar International Mobile Telecommunication-2000 (IMT-2000), que proporciona el marco para el acceso inalámbrico mundial juntando los distintos sistemas de redes terrestres y satelitales. 3G une una alta velocidad en el acceso móvil con servicios basados en Internet Protocol (IP) [2, 5].

3G incrementa la velocidad de transmisión de datos con tasas de hasta 2 Mbps para usuarios en interiores y de hasta 384 Kbps para usuarios peatonales. Esta generación opera en las frecuencias de 900 MHz y 2.1 GHz. Para la comunicación de voz y datos utiliza la comutación de paquetes. 3G aumenta la *Quality of Service (QoS)*, refuerza la seguridad de la información, mejora el rendimiento de los datos y de la transmisión de vídeos en tiempo real, restringe la interferencia de ruido y ofrece nuevos servicios como videollamadas y roaming global [2, 5]. Las principales desventajas de 3G se encuentran en que los planes de red son más caros y en que se requiere mayor energía en los teléfonos con respecto a los modelos de 2G [2].

3G contempla la introducción y empleo de estándares tecnológicos como *Code Division Multiple Access-2000 (CDMA-2000)*, la tecnología de acceso móvil *Wideband Code Division Multiple Access (WCDMA)*, *Universal Mobile Telecommunication System (UMTS)* basado en WCDMA, y *Time Division Synchronous Code Division Multiple Access (TD-SCDMA)*. 3G evoluciona hacia otras dos variantes: *High Speed Downlink Packet Access (HSDPA)* o 3.5G que mejora la velocidad de transmisión de datos de 8 Mbps a 10 Mbps en el enlace *downlink (DL)*, y *High Speed Uplink Packet Access (HSUPA)* o 3.75G que aumenta la velocidad del enlace *uplink (UL)* hasta los 5.8 Mbps [5, 6].

2.1.4. 4^ª Generación (4G)

En 2010 aparece la tecnología móvil 4^a Generación (4G), que es más confiable y está basada por completo en IP. El mayor progreso de 4G, en comparación con las anteriores generaciones tecnológicas, se centra en su mayor ancho de banda, que alcanza un máximo de 100 MHz mediante el uso de la técnica de agregación de portadoras de 20 MHz cada una. Con la banda ultra ancha ofrecida, la velocidad de

transferencia de datos en tiempo real se sitúa en los 100 Mbps en movimiento, y en 1 Gbps si existe poca movilidad, con un retraso inferior a 10 ms [5, 7].

Los servicios clave que brinda 4G son las videollamadas en *High Definition (HD)*, los SMSs, los MMSs, la transmisión de juegos en línea, la mejor QoS, la televisión móvil, el chat de vídeo, el roaming global, etc [2, 5]. Estos servicios de voz, datos y multimedia están disponibles para los usuarios a alta velocidad en cualquier lugar y en todos los instantes [2].

El *Third Generation Partnership Project (3GPP)* desarrolla el estándar de 4G *Long Term Evolution Advanced (LTE-A)* de comunicaciones inalámbricas para la transmisión de datos a alta velocidad para móviles [2]. LTE-A mejora la velocidad de transmisión de datos, la capacidad del sistema y la latencia, y trabaja en las bandas de 800 MHz, 900 MHz, 1800 MHz, 2100 MHz y 2600 MHz en Europa utilizando *Orthogonal Frequency Division Multiple Access (OFDMA)*, dividiendo el espectro en subcanales con espaciado de 15 KHz [6, 7].

En 4G también se desarrolla la tecnología *Worldwide Interoperability for Microwave Access (WiMAX)* de transmisión de datos utilizando ondas de radio [2]. Además, LTE-A implementa el protocolo de comunicación celular *Voice over Long Term Evolution (VoLTE)* de alta velocidad para realizar llamadas de voz en la red 4G [6].

2.1.5. 5ª Generación (5G)

Se denomina 5^a *Generación* (5*G*) a la tecnología de comunicaciones inalámbrica que viene a sustituir a 4G. 5G dispone de nuevas características como son una mayor velocidad de transmisión, un mayor ancho de banda, una baja latencia, un acceso centralizado controlado basado en la nube, una alta densidad de dispositivos conectados y una mayor seguridad y privacidad. La finalidad esencial de 5G es posibilitar la comunicación *machine to machine (M2M)* y la comunicación del *Internet of Things (IoT)* [1]. Como tecnología de acceso, 5G utiliza OFDMA [6].

El *Radiocommunication Sector of the International Telecommunication Union (ITU-R)* define los principales escenarios de uso de 5G en la Recomendación ITU-R M.2083-0, mostrados en la Figura 2.1 [8]:

- Enhanced Mobile Broadband (eMBB): En este caso de uso se pretende dar una velocidad elevada de datos, 10 a 100 veces mayor que las redes 4G [9]. eMBB potenciará nuevas aplicaciones como la realidad virtual, la realidad aumentada o el vídeo en HD en tiempo real, para ser usadas en la sanidad, la educación, etc. [1].
- Ultra Reliable Low Latency Communications (URLLC): Está dirigido hacia aplicaciones críticas que requieren baja latencia y conectividad ultra confiable con una tasa de error sumamente baja [9]. El control remoto de equipos sensibles, bajo circunstancias peligrosas, el control de técnicas médicas por remoto, etc. requerirán este tipo de comunicación, con una respuesta inmediata en tiempo real [1].
- Massive Machine Type Communications (mMTC): La aparición del IoT va a requerir la conexión de una enorme cantidad de dispositivos con alta conectividad entre sí, que funcionarán de forma autónoma y serán capaces de decidir por ellos mismos [1]. Se han elaborado estándares de conectividad para

una red que se pronostica densa y que va a requerir bajo consumo de energía en los dispositivos y baja latencia para la transmisión de datos [9].



Figura 2.1: Casos de uso 5G [10]

Los requisitos mínimos de 5G se recogen en la recomendación ITU-R M.2410. Se especifica la velocidad de datos pico, que se establece en 20 Gbps en el enlace DL y en 10 Gbps en el enlace UL. En referencia a las eficiencias espectrales máximas, presentan un valor de 30 b/s/Hz en el enlace DL y de 15 b/s/Hz en el enlace UL. En cuanto a la latencia del plano de usuario, ésta se fija en 4 ms para eMBB y en 1 ms para URLLC [11].

La arquitectura 5G tiene como objetivo procurar una eficaz transmisión inalámbrica de los datos, y evitar que se reduzca la eficiencia del espectro. Para lograr estos indicadores, 5G se sirve de avanzadas tecnologías que facilitan la comunicación de los usuarios en interiores y exteriores cuando se producen pérdidas en la misma, consiguiendo de esta manera una buena transmisión [12]. 5G se construye con tecnologías como *ondas milimétricas (mmWave)*, *Multiple Input Multiple Output (MIMO)* masivo, *Full-Duplex*, celdas pequeñas y *beamforming* [13].

Los dispositivos que son utilizados a diario funcionan en distintas frecuencias, la mayoría lo hacen por debajo de 6 GHz. El aumento de dispositivos y de usos provoca que los servicios ofrecidos sean más lentos y para solucionar este problema se opta por la tecnología mmWave, que aumenta de 6 GHz a 300 GHz el espectro [13]. Conforme al 3GPP, 5G ocupa determinadas bandas de frecuencia en el espectro divididas en dos rangos de frecuencia: *Frequency Range 1 (FR1)* y *Frequency Range 2 (FR2)*. El rango de frecuencia FR1 abarca desde los 4.1 GHz a los 7.125 GHz y se utiliza para conducir gran parte del tráfico de comunicaciones móviles, y el rango de frecuencia FR2 va desde los 24.25 GHz a los 52.6 GHz y está dedicado a obtener una mayor velocidad de los datos transmitidos, utilizándose en comunicaciones de corto alcance [14].

La tecnología MIMO masivo emplea grandes conjuntos de antenas para obtener una mejor *Signal to Noise Ratio (SNR)* por medio del *beamforming*, y así reducir la *Bit Error Rate (BER)* [14]. Mediante los haces formados, la energía de *radiofrecuencia (RF)* es focalizada unidireccionalmente para acrecentar la cobertura de la señal. La tecnología MIMO no precisa ocupar más espectro, amplía la capacidad de conexión de la red, y consigue una mayor tasa de datos y una mayor confiabilidad del enlace [12]. La tecnología de celda pequeña se utiliza para resolver el problema de la disminución de la intensidad de señal por causa de posibles obstáculos que se pueden interponer entre una torre de comunicación y un usuario. Consigue conexiones muy rápidas tanto en exteriores como en interiores [2, 13]. La tecnología *full-duplex* permite que los dispositivos inalámbricos puedan enviar y recibir datos a la vez y en la misma banda de frecuencias, para aumentar la capacidad del sistema, mejorar la latencia y proporcionar seguridad en la capa física [13].

El 3GPP define dos arquitecturas para la implementación de 5G [15]:

- Non-Stand Alone (NSA): En esta arquitectura la red de acceso radio 5G y su interfaz New Radio (NR) se emplean junto a la red core de infraestructura, Evolved Packet Core (EPC) y Long Term Evolution (LTE) ya existente, consiguiendo de esta manera que la tecnología NR se encuentre disponible sin necesidad de cambiar la red. Esta configuración solo soporta los servicios 4G pero utilizando las capacidades de 5G NR. NSA constituye un paso previo al despliegue completo 5G, y presenta la red de acceso 5G conectada a la red core 4G. La arquitectura NSA se muestra en la Figura 2.2, donde se observa que la BS NR (5G), también denominada nodo lógico "en-gNodeB (gNB)", se conecta a la BS LTE (4G), o también conocida como nodo lógico "eNodeB (eNB)", a través de la interfaz X2. Esta interfaz también posibilita la conexión de dos eNBs. La arquitectura NSA toma la conexión de la BS 4G como nodo primario y de la BS 5G como nodo secundario, dando como resultado una conectividad dual Evolved Universal Terrestrial Radio Access (E-UTRA)-NR a través de la red de acceso 5G (NR) y 4G (E-UTRA).
- Stand Alone (SA): Esta arquitectura se caracteriza por la conexión de la interfaz NR a la red core 5G. Además, esta configuración soporta el conjunto completo de servicios 5G. SA constituye el despliegue completo de 5G debido a que no necesita la red 4G para funcionar. La arquitectura SA se muestra en la Figura 2.3, donde se aprecia que la interfaz Xn permite la conexión de las BSs NR, también denominadas nodo lógico "gNB", entre ellas. Además, la interfaz NG posibilita la comunicación entre la red de acceso, también conocida como Next Generation Radio Access Network (NG-RAN) para SA, y la red core 5G.



Figura 2.2: Arquitectura NSA [7]



Figura 2.3: Arquitectura SA [7]

2.2. Arquitectura C-RAN

La arquitectura *Distributed Radio Access Network (D-RAN)* se considera la solución precursora a *Cloud Radio Access Network (C-RAN)*, pues introduce la idea de dividir la estación base, o BS, en dos elementos: *Baseband Unit (BBU)* y *Remote Radio Head (RRH)*. La unidad BBU se encarga de realizar las funciones de procesado en banda base y la unidad RRH se centra en las funciones radio, estando situada esta unidad junto a las antenas en la torre [3].

Los elementos BBU y RRH se conectan mediante un segmento de red denominado *fronthaul*, en el que las señales de radio se transmiten mediante el protocolo *Common Public Radio Interface (CPRI)* [3]. El *fronthaul* se despliega haciendo uso de fibra óptica o mediante la radiocomunicación por microondas. El *backhaul* define el enlace existente entre la red *core* y la BBU, transmitiendo por él, típicamente a través de fibra óptica, los datos de control, de gestión y de los usuarios [16]. En la Figura 2.4 se muestra la arquitectura D-RAN descrita.



Figura 2.4: Arquitectura D-RAN [16]

La arquitectura D-RAN constituye una solución eficiente para las redes celulares 3G y 4G. Sin embargo, no se trata de una solución adecuada para las redes 5G, debido a que no es suficientemente

escalable y a que no cumple con los requisitos de baja latencia y alto ancho de banda establecidos por estas redes [16].

C-RAN constituye una arquitectura centralizada fundamentada en la virtualización en tiempo real y en la computación en la nube, con el objetivo de proveer servicios fiables para las redes de acceso inalámbricas de la próxima generación [17]. C-RAN se basa en la separación de las BBUs con respecto a sus correspondientes RRHs, agrupando las BBUs en el denominado BBU *pool*. Cada elemento RRH se conecta a su correspondiente BBU *pool* mediante el enlace *fronthaul*. El enlace *backhaul* comunica cada BBU *pool* con la red *core* [16]. En la Figura 2.5 se muestra la arquitectura C-RAN comentada.



Figura 2.5: Arquitectura C-RAN [3]

A modo de resumen, a continuación, se describen los elementos que conforman la arquitectura C-RAN [18]:

- BBU *pool*: La arquitectura C-RAN migra el procesado en banda base a una zona centralizada, conocida como BBU *pool*, con el objetivo de obtener una mejor gestión y un procesado más adecuado. El BBU *pool* garantiza una gestión de la señal y un procesado en banda base centralizados [18]. El BBU *pool* está formado por BBUs que funcionan como estaciones base virtuales para optimizar el reparto de los recursos radio y para procesar las señales en banda base [19].
- RRH: Estos elementos se encargan de efectuar la conversión analógica-digital y digital-analógica, llevan a cabo el procesamiento digital, y realizan el filtrado y la amplificación de potencia de las señales de RF [18].
- Fronthaul: Consiste en un enlace de baja latencia y alta capacidad que conecta los RRHs con el BBU pool [19]. El fronthaul permite el transporte de la información de control y de los datos [18]. Este enlace se puede llevar a cabo mediante diversas tecnologías, tanto comunicaciones inalámbricas como por fibra óptica [19].
- Backhaul: Se denomina backhaul al enlace que interconecta el BBU pool con la red core [18].

La arquitectura C-RAN permite aumentar la escalabilidad de la red, mejorar el *throughput* de la misma y su eficiencia espectral, reducir el consumo de energía, simplificar tanto la gestión como el mantenimiento

de la red y facilitar el balanceo de carga [16]. Por otro lado, C-RAN consigue reducir los costes con respecto a las redes móviles convencionales, debido a que las BBUs se centralizan en un solo sitio [20]. La virtualización de la BBU *pool* propicia el procesado compartido, es decir, permite compartir los recursos de procesado que se encuentren disponibles entre diferentes sitios y posibilita la asignación de recursos adicionales cuando sean necesarios en diversas áreas [3].

A pesar de las numerosas ventajas de la arquitectura C-RAN, existe un gran problema: precisa de una red *fronthaul* de baja latencia y alta capacidad que comunique las localizaciones remotas con el centro de datos. Esto provoca que los operadores de red asuman altos costes debido a que no pueden reutilizar sus redes con menos capacidades ya existentes [21]. Para solucionar este problema surge el concepto de *functional split*: no se centralizan todas las funciones de procesamiento de una BS, sino que se selecciona un conjunto de ellas para ser implementadas en una unidad central y el resto de funciones se despliegan en una unidad distribuida [22].

La introducción del concepto de *functional split* causa un cambio de arquitectura, donde la BBU se divide en los elementos *Centralized Unit (CU)* y *Distributed Unit (DU)*, y el RRH pasa a denominarse *Remote Radio Unit (RRU)* [3, 4]. En la Figura 2.6 se muestra la arquitectura C-RAN con la implementación del *functional split*.



Figura 2.6: Arquitectura C-RAN con la implementación del functional split [4]

A continuación, se describen brevemente los principales elementos que conforman la nueva arquitectura C-RAN con división funcional [4]:

- CU: Unidad centralizada que se implementa en una localización central y que agrega las funciones de las capas superiores de las estaciones base [4, 23]. Las funciones asignadas a la CU se benefician de una alta potencia de procesado en el centro de datos y de una centralización del procesamiento [3]. La CU puede controlar una o varias DU y se comunica con ellas mediante el enlace *midhaul*. Las funciones exactas de la CU vienen determinadas por el *functional split* [4].
- DU: Unidad distribuida que lleva a cabo funciones próximas a los usuarios debido a que se sitúa cerca de los equipos radio [3, 23]. La DU implementa funcionalidades de capa física y algunas funciones de

las capas superiores. Está controlada por una CU y sus funciones concretas dependen del *functional split* implementado [4].

RRU: Esta unidad está formada por todos los componentes necesarios para la transmisión y la recepción de las señales [4]. Se encarga de realizar las funciones radio y parte de las funcionalidades de la capa física [24]. Se comunica con la DU mediante el enlace *fronthaul* [4].

2.3. Functional split

Como se ha comentado en la sección anterior, las funciones de una BS se dividen entre la CU y la DU. El concepto de *functional split* permite decidir la cantidad de funcionalidades que se localizan en la zona de la antena y el número de funciones que se centralizan en un centro de procesamiento de datos con gran capacidad [3].

A continuación, se definen los protocolos que conforman la pila de protocolos de una BS LTE, mostrados en la Figura 2.7, con el objetivo de especificar cuáles son las funciones concretas de una BS [25]. Cabe destacar que, aunque para esta descripción se usará la pila de protocolos de LTE, la pila para 5G definida por el 3GPP es similar y las divisiones funcionales identificadas son las mismas.



Figura 2.7: Pila de protocolos de una BS LTE [26]

La pila de protocolos de una BS LTE está conformada, para el plano de datos, por los siguientes protocolos: *Physical (PHY), Medium Access Control (MAC), Radio Link Control (RLC)* y *Packet Data Control Protocol (PDCP)*. Para el caso del plano de control, se añade a los protocolos comentados el protocolo *Radio Resource Control (RRC)* [25].

El protocolo PHY se encarga de gestionar la conectividad, es decir, la modulación, codificación, etc, entre la BS y el equipo de usuario o *User Equipment (UE)* a través del aire. Este protocolo proporciona canales de transporte hacia el protocolo MAC. Éste último se ocupa de la planificación de los enlaces DL y UL, realiza el mapeo entre los canales de transporte y los canales lógicos hacia RLC, y participa en el multiplexado y demultiplexado de información [25, 26].

RLC gestiona la entrega de *Protocol Data Units (PDU)*s de tráfico entre la BS y el UE. El protocolo RLC permite realizar la entrega haciendo uso de tres esquemas diferentes: modo transparente, modo no reconocido y modo reconocido. El protocolo PDCP se encarga de la comprobación de la integridad de los

datos de los usuarios, así como del cifrado de los mismos. PDCP también gestiona la compresión de la cabecera IP con el objetivo de optimizar la utilización del ancho de banda [25, 26].

El protocolo RRC se sitúa encima del protocolo PDCP y se ocupa de controlar la portadora radio, de gestionar las medidas del UE y de la difusión de información del sistema. RRC lleva a cabo la gestión de la conexión, además de realizar la administración de la movilidad del UE en traspasos dentro de LTE [25, 26].

Los principales beneficios que presentan las arquitecturas que implementan flexibilidad en la separación de funciones entre la CU y la DU se recogen a continuación [27]:

- Las implementaciones flexibles de elementos *hardware* permiten conseguir soluciones escalables y rentables económicamente.
- La arquitectura de separación de funciones entre la CU y la DU facilita la gestión de la carga y la coordinación del sistema, permite una optimización del rendimiento en tiempo real y posibilita la utilización de soluciones Software Defined Network (SDN)/Network Function Virtualization (NFV).
- La adaptación a diferentes casos de uso, como una latencia cambiante, es posible gracias a los *functional splits* configurables.

2.3.1. Descripción de los functional splits posibles entre la CU y la DU

La decisión de cómo separar las funciones de la BS depende de diversos factores relacionados con los servicios soportados que se quieren proveer, con las restricciones existentes y con los escenarios de despliegue de redes. Existen tres principales factores a tener en cuenta, que se describen a continuación. El primero es la necesidad de proporcionar una determinada QoS para cada servicio, garantizando, por ejemplo, un alto *throughput*, una baja latencia, etc. El segundo se refiere a la necesidad de satisfacer, para cada zona geográfica, una determinada demanda de carga y soportar una densidad de usuarios concreta. El tercero se centra en la necesidad de funcionar de forma óptima con redes de transporte que presenten diversos niveles de rendimiento [27].

Basándose en la pila de protocolos de una BS LTE, explicada anteriormente, el 3GPP define en el *Technical Report* 38.801 ocho *functional splits* posibles entre la CU y la DU, como se muestra en la Figura 2.8. Seguidamente, se realiza una descripción detallada de cada uno de los *functional splits* posibles indicando sus beneficios potenciales [27].



Figura 2.8: Functional splits posibles entre la CU y la DU definidos por el 3GPP [27]

Opción 1: RRC/PDCP

Este *functional split* sitúa el protocolo RRC en la CU y localiza a los protocolos PDCP, RLC, MAC, PHY y RF en la DU. Con esta separación, el plano de datos completo se encuentra localizado en la DU [27].

Esta configuración proporciona beneficios a la hora de manejar casos de uso de latencia baja o de *Edge Computing*, donde la información de los usuarios debe situarse cerca del punto de transmisión [27].

El rendimiento de este *functional split* se puede ver afectado por la securización de la interfaz en despliegues de tipo prácticos, debido a la separación de los protocolos RRC y PDCP [27].

Opción 2: PDCP/RLC

Esta opción se divide en dos alternativas: la Opción 2-1 y la Opción 2-2. En la primera, se sitúa a los protocolos RRC y PDCP en la CU y a los protocolos RLC, MAC, PHY y RF en la DU. Este *functional split* permite centralizar la agregación de tráfico de los puntos de transmisión NR y E-UTRA. También, favorece la gestión de la carga de tráfico existente entre los puntos de transmisión NR y E-UTRA [27].

Esta opción de *split* constituye la más sencilla de estandarizar, debido a que sus fundamentos ya se han definido para la conectividad dual de LTE. Por otro lado, este *functional split* mejora la retransmisión de PDUs RLC perdidas [27].

La Opción 2-2 se caracteriza por la separación de PDCP y RRC en entidades centrales diferentes. De esta manera, se consigue aislar el plano de datos del plano de control en la CU. Al igual que ocurría en la Opción 2-1, esta alternativa permite la centralización de la agregación de tráfico de los puntos de transmisión NR y E-UTRA, así como la gestión de la carga de tráfico entre ellos [27].

Esta opción facilita la centralización del protocolo PDCP, además de permitir tener un plano de datos separado y el protocolo RRC centralizado. Sin embargo, es necesario asegurar la coordinación de las configuraciones de seguridad entre diversas instancias de PDCP [27].

Opción 3: High RLC/Low RLC

Esta opción presenta dos aproximaciones diferentes: una basada en una separación de funciones que operan en tiempo real y la otra en una separación de funciones que se ejecutan en tiempo no real [27].

La Opción 3-1 se fundamenta en una separación basada en *Automatic Repeat Request (ARQ)*. Este *functional split* divide el protocolo RLC en dos subcapas: RLC alto y RLC bajo. RLC alto está situado en la CU y se encarga de ARQ y de llevar a cabo todas las funciones RLC. RLC bajo se localiza en la DU y se ocupa de realizar la segmentación [27].

Este *split* permite centralizar la agregación de tráfico de los puntos de transmisión NR y E-UTRA además de gestionar la carga de tráfico entre ellos. Otro beneficio que presenta es un mejor control de flujo en el *split*. Además, la localización de ARQ en el CU proporciona ganancias de centralización. La señalización del plano de control y la información crítica se encuentran protegidos frente a fallos de la red de transporte gracias al mecanismo ARQ de la CU [27].

En esta opción, las DUs manejan más UEs conectados, debido a que no se necesita el contexto del UE y no existen datos de estado RLC almacenados. Este *split* se considera el más robusto en condiciones de transporte que no son ideales, ya que el ordenamiento de los paquetes y ARQ se llevan a cabo en la CU.

La centralización de ARQ conlleva la reducción de los requisitos de almacenamiento y de procesado en la DU [27].

Sin embargo, este *functional split* presenta mayor sensibilidad a la latencia que el *split* que asigna ARQ a la DU, debido a que, en redes de transporte con separación de funciones, las retransmisiones son susceptibles a la latencia de dichas redes [27].

La Opción 3-2 se basa en un *split* fundamentado en la transmisión y la recepción RLC. Esta opción separa el protocolo RLC en las subcapas RLC alto y RLC bajo, como ocurría en la Opción 3-1. RLC bajo se encarga de la transmisión del DL y de la recepción del UL y RLC alto se ocupa de la recepción del DL y de la transmisión del UL. RLC bajo se localiza en la DU y RLC alto se sitúa en la CU [27].

Este *functional split* no es sensible a la latencia existente en la red de transmisión entre la DU y la CU. Permite centralizar la agregación de tráfico de los puntos de transmisión NR y E-UTRA, además de gestionar la carga de tráfico entre ellos. El control de flujo se lleva a cabo en la CU y requiere de un *buffer* en esta unidad. En la DU se sitúa el *buffer* de transmisión, de forma que el tráfico con control de flujo proveniente de la CU se pueda almacenar antes de su transmisión [27].

En comparación con el caso en el que no se divide el protocolo RLC, las PDUs de estado pueden ocasionar un retraso temporal adicional ya que deben ser enviadas de la CU a la DU antes de ser transmitidas por el aire. Otra desventaja consiste en la necesidad de dos *buffers* para la transmisión, uno en la CU y otro en la DU, para realizar el control de flujo en la CU y la transmisión RLC en la DU [27].

Opción 4: RLC/MAC

Este *functional split* localiza en la CU a los protocolos RRC, PDCP y RLC mientras que sitúa en la DU a los protocolos MAC, PHY y RF. Esta opción de *split* no presenta beneficios para el caso de uso de la pila de protocolos de LTE mientras que, para el caso de la pila de protocolos de NR, sus posibles beneficios se están analizando [27].

Opción 5: intra MAC

Esta opción de *functional split* se basa en la localización de los protocolos PHY, RF y MAC bajo en la DU y de los protocolos MAC alto, RLC, RRC y PDCP en la CU. De esta forma, la capa MAC se divide en dos subcapas, MAC alto y MAC bajo, situándose sus funciones y servicios en la CU, en la DU o en ambas [27].

La subcapa MAC alto está formada por un planificador centralizado que se encarga de controlar varias subcapas MAC bajo, lo que conlleva una decisión de la planificación centralizada de alto nivel. Además, MAC alto permite adoptar diversos métodos de coordinación de la interferencia entre células, como *Coordinated MultiPoint (CoMP)* [27].

La subcapa MAC bajo presenta funcionalidades críticas en el tiempo como funciones con requisitos de retardo exigentes o funciones cuyo rendimiento es proporcional a la latencia. Las funciones específicas radio en MAC bajo se encargan de procesar los datos relacionados con la planificación [27].

Este *functional split* permite centralizar la agregación de tráfico de los puntos de transmisión E-UTRA y NR, además de gestionar la carga de tráfico entre ellos. Esta opción facilita la reducción del ancho de banda necesario en el enlace *fronthaul* y minimiza los requisitos de latencia en dicho enlace. Por otro lado, permite la gestión de la interferencia de forma eficiente entre múltiples células, así como la mejora de las

tecnologías de planificación [27].

Sin embargo, este *split* introduce complejidad en el interfaz CU-DU y añade dificultad en la definición de las operaciones de planificación sobre DU y CU. Por otro lado, la decisión de planificación entre ambas unidades se ve afectada por los retrasos en el *fronthaul* [27].

Opción 6: MAC/PHY

Esta opción de *split* sitúa al protocolo MAC y a las capas superiores en la CU, mientras que localiza en la DU a las capas PHY y RF. La interfaz existente entre la CU y todas las DUs se encarga de transportar información relacionada con la planificación, datos, configuración y medidas [27].

Este *functional split* permite centralizar la agregación de tráfico de los puntos de transmisión E-UTRA y NR, así como gestionar la carga de tráfico entre ellos. Además, minimiza los requisitos en el *fronthaul* en relación al *throughput*. Debido a que la capa MAC se encuentra en la CU, es posible realizar una transmisión conjunta y una planificación centralizada. Esta opción permite llevar a cabo un *pool* de recursos para la capa MAC y las capas superiores [27].

Sin embargo, este *split* precisa de interacciones temporales a nivel de subtrama entre las capas PHY de las DUs y la capa MAC de la CU. Por otro lado, el *Round Trip Time (RTT)* del enlace *fronthaul* afecta a la planificación y a la sincronización del *Hybrid Automatic Repeat Request (HARQ)* [27].

Opción 7: intra PHY

Esta opción de *split* localiza la parte superior de la capa PHY y las capas superiores en la CU, mientras que en la DU se sitúa la parte inferior del protocolo PHY y la capa RF. Esta configuración consta de tres subopciones: Opción 7-1, Opción 7-2 y Opción 7-3. Además, permite seleccionar opciones diferentes para los enlaces DL y UL. Por otro lado, el ancho de banda de transporte necesario entre la CU y la DU se puede minimizar con una técnica de compresión [27].

Este *functional split* permite centralizar la agregación de tráfico de los puntos de transmisión NR y E-UTRA, así como gestionar la carga de tráfico entre ellos. Las tres sub-opciones existentes minimizan los requisitos del enlace *fronthaul* en relación al *throughput*. Debido a que el protocolo MAC se encuentra centralizado, es posible realizar una planificación y un procesado conjuntos. Sin embargo, este *split* precisa de interacciones temporales a nivel de subtrama entre la parte de la capa PHY situada en la CU y la parte de la capa PHY localizada en las DUs [27].

La Opción 7-1 propone localizar en la DU las funciones de filtrado *Physical Random Access Channel* (*PRACH*). Esta configuración permite implementar receptores avanzados. La Opción 7-2 sitúa en la DU, para el caso del UL, las funciones de prefiltrado y la des-asignación de recursos, y, para el caso del DL, localiza en la DU las funciones de precodificación y la asignación de recursos. La Opción 7-3, solo disponible para el DL, localiza en la CU al codificador y sitúa en la DU el resto de funciones de la capa PHY. Esta configuración minimiza los requisitos del enlace *fronthaul* en relación con el *throughput* [27].

Opción 8: PHY/RF

Este *split* separa al protocolo PHY de la parte RF, por lo que se consigue la centralización de procesos de todas las capas de protocolos, excepto RF, que se lleva a cabo en la DU, lo que provoca una coordinación estrecha de la *Radio Access Network (RAN)*. Esto supone un apoyo eficiente de funcionalidades como la movilidad, el balanceo de carga, MIMO y CoMP [27].

Este *functional split* permite centralizar la agregación de tráfico de los puntos de transmisión NR y E-UTRA, así como gestionar la carga de tráfico entre ellos. Proporciona además altos niveles de coordinación y de centralización de toda la pila de protocolos, lo que conlleva un rendimiento radio y una gestión de los recursos más eficientes [27].

La separación de las capas PHY y RF habilita el aislamiento de los componentes RF con respecto a las actualizaciones de la capa PHY, lo que produce una mejora de la escalabilidad RF/PHY. Esta división facilita la reutilización de los componentes RF para servir a los protocolos PHY de diversas tecnologías de acceso radio, como GSM, 3G o LTE. También permite realizar un *pool* de los recursos PHY, lo que posibilita un dimensionamiento de la capa PHY más eficiente en términos económicos. Además, los operadores pueden compartir los componentes RF gracias a este *functional split*, reduciendo el coste de los sistemas y de las localizaciones [27].

Sin embargo, esta opción de *split* conlleva altos requisitos en la latencia del enlace *fronthaul*, lo que implica restricciones en los despliegues de red en relación con las opciones de transporte disponibles y con la topología de red. Otra desventaja de este *split* es que existen requisitos exigentes con respecto al ancho de banda del *fronthaul*, lo que provoca costes en el dimensionamiento del transporte y un consumo de recursos mayor [27].

2.4. Network slicing

Las redes 5G necesitan integrar diversos servicios con diferentes requisitos de rendimiento (baja latencia, alta movilidad, alta fiabilidad, alta seguridad y alto *throughput*) en una infraestructura de red física única, y proporcionar a cada servicio una red lógica personalizada [28]. El concepto de *Network slicing* se refiere a la división de una red física en múltiples redes virtuales, siendo cada una de ellas optimizada y diseñada para un servicio concreto [29]. El *Network slicing* constituye un modelo *Network as a Service (NaaS)* y permite asignar y reasignar recursos de forma flexible en función de las demandas dinámicas, de forma que es capaz de personalizar los *slices* de red para escenarios 5G diversos y complejos [28].

Un *slice* de red consiste en una red virtual creada encima de una red física de forma que parece que funciona en una red física dedicada. Se trata de una red autocontenida con su propia topología, sus reglas de aprovisionamiento, sus recursos virtuales y su flujo de tráfico. Los *slices* se encuentran aislados lógicamente pero se pueden compartir los recursos entre ellos. Para cumplir los requisitos de los servicios, el *network slicing* se implementa de extremo a extremo. El *network slicing* de 5G incorpora la división de la red *core* 5G, de la RAN 5G y de los dispositivos de usuario final [28].

La implementación del *network slicing* se basa en dos herramientas fundamentales: NFV y SDN. La tecnología NFV provee las *Virtual Network Functions (VNF)* necesarias para el plano de datos de los *slices* de la red *core*, de acuerdo con el tipo de servicio [29]. A medida que el servicio varía de forma dinámica, estas VNFs se pueden escalar bajo demanda [28]. La tecnología SDN se encarga de separar el plano de control y el plano de datos de la red *core* con el objetivo de conseguir un despliegue independiente del tráfico [29]. Las tecnologías NFV y SDN permiten controlar los *slices* y modificarlos cuando se considere necesario, sin tener en cuenta la infraestructura física de la red [30].

El Network slicing personaliza cada uno de los slices para cada servicio, asignando la cantidad correcta

de recursos a cada *slice*. El servicio vinculado con cada *slice* especifica el número de recursos necesarios para dicho *slice*. El *Network slicing* se lleva a cabo en función de las características necesarias de cada *slice*, las cuales se recogen a continuación [30]:

- Aislamiento: Los *slices* de red son capaces de aislar los recursos de red de un servicio concreto de los demás. Las configuraciones entre *slices* están aisladas entre sí. De esta forma, se mejora la seguridad y la fiabilidad de cada *slice* [28].
- Personalización: Un *slice* de red se personaliza en función de los requisitos del servicio, lo que conlleva la optimización de la asignación de los recursos de red físicos [28]. Esta personalización ocasiona que los usuarios obtengan los servicios con un nivel elevado de calidad, lo que permite el aumento de los ingresos de los operadores [30].
- Programabilidad: Esta características facilita que los usuarios tengan control sobre los recursos disponibles en sus correspondientes *slices* [30].
- Automatización: Permite que el *Network slicing* se realice de forma automática, sin necesidad de intervención manual. Esto se lleva a cabo gracias a la información que envía el usuario sobre la latencia, la capacidad y la información de planificación de la partición [30].
- Flexibilidad: A medida que cambia el número de usuarios y los requisitos del servicio, el *slice* de red se puede escalar hacia arriba o hacia abajo [28].

Cada uno de los *slices* generados por el *Network slicing* integra un caso de uso concreto que soporta diferentes verticales, como la gestión del hogar, la seguridad, la cirugía remota, la infraestructura automotriz, etc, como se muestra en la Figura 2.9. Cuando ocurre una imprecisión o un fallo en un *slice*, no repercute en los demás *slices*, debido a que están aislados unos de otros [31].



Figura 2.9: Casos de uso de los slices obtenidos gracias al Network slicing [31]

La arquitectura del *Network slicing*, mostrada en la Figura 2.10, presenta cuatro capas fundamentales: capa de infraestructura, capa de función de red, capa de servicio y capa de gestión y orquestación. A continuación, se presenta una descripción de las capas comentadas y se especifican sus funciones concretas [31]:

- Capa de infraestructura: Esta capa se corresponde con la infraestructura de red física que extiende la RAN, la red *core* y la *edge/cloud*. También incluye el despliegue de la infraestructura, la gestión, el control y la asignación de recursos a los *slices*, de manera que las capas superiores manejan estos recursos e informan sobre ellos.
- Capa de función de red: Lleva a cabo todas las funcionalidades y las operaciones relativas a la configuración de las funciones de red. Las tecnologías NFV y SDN se consideran un aspecto técnico importante en esta capa. Además, facilita la ubicación de los *slices* de red en el encadenamiento de múltiples *slices*. Esta capa gestiona de forma eficiente las funciones de red de grano fino y de grano grueso.
- Capa de servicio: Está formada por vehículos conectados, dispositivos móviles y sistemas de realidad virtual. Esta capa se encuentra relacionada de forma directa con diversos modelos de negocio y simboliza la función de red y las expectativas de funcionamiento de la infraestructura de red. Los VNFs se mapean en recursos físicos en base a la definición de los servicios de forma que no se incumpla el *Service Level Agreement (SLA)*.
- Capa de gestión y orquestación: Se trata de una capa transversal que presenta tres funcionalidades clave. La primera se basa en la utilización de las funciones de la capa de infraestructura para generar *Virtual Network Instances (VNI)* en la red física. La segunda consiste en la creación de una cadena de servicio gracias a la agregación de las funciones de red y de la capa de virtualización mediante el mapeo de las funciones de red a VNI. La tercera se encarga de mantener la comunicación entre la estructura del *Network slicing* y el servicio para adecuar los recursos virtualizados a los cambios, y para gestionar el ciclo de vida del VNI.



Figura 2.10: Arquitectura del Network slicing [31]

Capítulo 3

Planteamiento del problema

En este capítulo se realiza una descripción detallada del problema a resolver, así como de los escenarios de aplicación que se han considerado. Además, se especifican los requisitos teóricos del enlace *fronthaul* para cada *functional split* en términos de latencia y *throughput*, y se introduce el concepto de complejidad computacional.

3.1. Requisitos del enlace fronthaul para cada split

El *Small Cell Forum* define los requisitos teóricos que debe cumplir el enlace entre la CU y la DU para garantizar el correcto funcionamiento de cada *functional split*. Éstos se proporcionan en términos de latencia y de *throughput* [32].

En relación al *throughput*, se presentan, para cada *functional split*, los valores mínimos a cumplir para los enlaces UL y DL, con el objetivo de garantizar un funcionamiento adecuado del enlace *fronthaul*, tanto en el plano de control como en el de datos [32].

Con respecto a la latencia, se definen cuatro grupos, considerando únicamente un solo sentido de la comunicación: Ideal, Near Ideal, Sub Ideal y Non Ideal. La latencia de tipo Ideal presenta un valor de 250 μ s, la Near Ideal de 2 ms, la Sub Ideal de 6 ms y la Non Ideal de 30 ms [32].

Para cada opción de *functional split*, se estudia el impacto de presentar los cuatro tipos de latencia definidos. Éste puede ser de tipo N/A, donde se considera que el *split* no es realizable con ese valor de latencia, de tipo despreciable, donde el *split* es compatible con esa latencia, o de tipo descriptivo, en situaciones en las que se puede producir una degradación del rendimiento debido a esa latencia, y se describe esa posible limitación [32].

A continuación, para cada *functional split*, se especifican los valores mínimos de *throughput* necesarios para conseguir un funcionamiento correcto del sistema, y se describe el impacto de los niveles de latencia en cada *split* [32].

Para el caso del *split* RRC/PDCP, se considera que el impacto de la latencia es de tipo despreciable para los grupos Ideal, Near Ideal y Sub Ideal. Para la situación Non Ideal, algunos indicadores de rendimiento fundamentales pueden ser degradados, debido al retraso adicional en la gestión de procedimientos RRC. El *throughput* del *fronthaul* establecido para el DL es de 187.5 Mbps y el del UL es de 62.5 Mbps. Además,

se determina que el throughput destinado al control es de 10 Mbps [32].

En cuanto al *split* PDCP/RLC, el impacto de la latencia se asume como despreciable para los casos Ideal, Near Ideal y Sub Ideal. Para el grupo Non Ideal, la latencia provoca la degradación de la calidad de servicio en aplicaciones como los juegos en línea. Como ocurría en el caso anterior, en este split el *throughput* mínimo requerido para el DL es de 187.5 Mbps y el del UL es de 62.5 Mbps [32].

En relación al caso del *functional split* RLC/MAC, los requisitos del *fronthaul* dependen de la implementación de un esquema de control de flujo. Si no se utiliza, las necesidades en cuanto a la latencia serían más estrictas, precisando de una latencia ideal de 250 μ s. En el caso de implementar un esquema de control de flujo, se considera que la latencia presenta un impacto despreciable para los grupos Ideal, Near Ideal y Sub Ideal, siendo capaz el sistema de funcionar adecuadamente hasta un retardo de 30 ms. El *throughput* mínimo para el DL es de 187.5 Mbps y para el UL de 62.5 Mbps, al igual que en los casos anteriores [32].

Con respecto al *split* intra MAC, para que su funcionamiento sea admisible en un enlace *fronthaul* no ideal, el planificador MAC debe ser capaz de enviar decisiones con antelación, antes de que sean necesarias. En la Tabla 3.1, se observa, para cada grupo de latencia, el tiempo de respuesta HARQ y el número máximo de retransmisiones HARQ requeridos. Se aprecia que, a medida que aumenta la latencia, también lo hace el tiempo de respuesta y el número máximo de retransmisiones. Adicionalmente, al incrementarse este último parámetro, se reduce el rendimiento del sistema. En cuanto al *throughput*, se requieren 187.5 Mbps en el DL y 62.5 Mbps en el UL, como en casos anteriores [32].

Tipo	Tiempo respuesta	Número máximo
latencia	HARQ (ms)	retransmisiones HARQ
Ideal	8	0
Near Ideal	12	1
Sub Ideal	20	2
Non Ideal	68	8

Tabla 3.1: Requisitos del *split* intra MAC en términos de latencia [32]

Para el caso del *split* MAC/PHY, se definen también, para cada tipo de latencia, el tiempo de respuesta HARQ y el número máximo de retransmisiones HARQ. Cabe destacar que, para el caso de latencia Non Ideal, este *functional split* no es realizable. Los valores del resto de grupos de latencia, en términos de tiempo de respuesta y número de retransmisiones HARQ, coinciden con los definidos en el *split* anterior. Como ocurre en los casos previos, el *throughput* mínimo del DL se establece en 187.5 Mbps y el del UL en 62.5 Mbps [32].

En cuanto al *split* intra PHY, presenta los mismos requisitos, en función de la latencia y en términos de tiempo de respuesta y número de retransmisiones HARQ, que en el caso del *split* MAC/PHY. Igualmente, no es realizable en escenarios cuya latencia sea de tipo Non Ideal. Con respecto al *throughput*, se especifican diferentes valores mínimos en función de la cantidad de funciones que se lleven a cabo en la CU, siendo mayor cuantas más funciones se centralicen. Por lo tanto, el *throughput* mínimo para el DL varía desde los 173 Mbps hasta los 2457.6 Mbps, y para el UL abarca desde los 452 Mbps hasta los 2457.6 Mbps [32].

A diferencia de los requisitos teóricos de throughput y latencia establecidos para cada functional
split, no existen valores ampliamente aceptados de la complejidad computacional, debido a que este parámetro depende en gran medida de la implementación de la BSs y de su configuración. En la literatura, se determina que la complejidad varía en función de la centralización, presentando cada *functional split* un valor. Por lo tanto, cuantas más funciones se centralicen, mayor complejidad computacional existirá en la CU. Además, esta complejidad computacional de las operaciones de procesado de señal se expresa en términos de Giga Operaciones por Segundo (GOPS) [33-36].

En este trabajo, se ha obtenido el porcentaje de la complejidad computacional total de una BS que se corresponde con cada uno de los *functional splits*, basándose en los datos de la Tabla IV en [33]. Aplicando estos datos, mostrados en la Tabla 3.2, y un valor arbitrario de la complejidad computacional total de la BS, se obtiene el valor correspondiente para cada *split*. En este caso, se supone que la BS presenta una complejidad computacional total en la CU de 20 GOPS, por lo que los valores resultantes de complejidad para cada *functional split* se corresponden con los mostrados en la Tabla 3.2.

Functional split	Ratio	Complejidad computacional	
C-RAN	0.96	20.9	
Intra PHY	1.25	16	
MAC/PHY	1.87	10.7	
Intra MAC	2.3	8.7	
RLC/MAC	2.99	6.7	
Intra RLC	3.51	5.7	
PDCP/RLC	4.26	4.7	
RRC/PDCP	7.41	2.7	

Tabla 3.2: Ratios y complejidad computacional de cada *functional split* [33]

3.2. Descripción del problema

Se considera un escenario donde se despliegan servicios de varios verticales y donde la conectividad es proporcionada por BSs físicas mediante *slices* de red. Para ello, los recursos físicos de las BSs se distribuyen entre los *slices*, los cuales pueden cubrir varias áreas. Cada *slice* debe proporcionar en cada área donde está presente un *throughput* mínimo y un *throughput* objetivo. Como se describe en [22], las diferentes opciones de centralización permiten utilizar diversas técnicas de coordinación, lo que mejora la *Signal to Interference plus Noise Ratio (SINR)* recibida por los usuarios finales.

A modo de ejemplo, en la Figura 3.1 se muestra un escenario formado por un solo área, cubierta por una BS física, donde se debe proporcionar un *throughput* mínimo y un *throughput* objetivo por cada *slice* desplegado. Además, se debe seleccionar el nivel de *split* de forma que la capacidad computacional de la CU no se sature.

Teniendo esto en cuenta, se pretende conseguir una solución que se adecúe a los cambios en las condiciones inalámbricas, mediante la adaptación tanto del nivel de centralización como de la distribución de los recursos físicos entre los *slices*. La formulación del problema que se va a presentar es genérica, pero se considera que la escala del tiempo de decisión es lo suficientemente grande para despreciar el tiempo transcurrido en la reconfiguración de la red.



Figura 3.1: Escenario de referencia

3.2.1. Modelo del sistema

Se define \mathcal{B} como el conjunto de BSs físicas y \mathcal{S} como el conjunto de *slices* o servicios. Cada BS $b \in \mathcal{B}$ presenta un número de recursos físicos inalámbricos η_b y cada *slice s* tiene una demanda de *throughput* d_s , medida en unidades de tasa de tráfico. Por lo tanto, el objetivo es satisfacer dicha demanda en media a lo largo del tiempo.

Para satisfacer la demanda de los *slices*, se instancian BSs virtuales dentro de las físicas. El tiempo se ranura y la variable *t* indica un *slot* determinado. Para cada *slot* se decide la cantidad de recursos asignados a cada BS virtual, y se determina el nivel de centralización.

La demanda de capacidad se debe cumplir en todo el área cubierta por la BS física. Además, las áreas solapadas se consideran de forma independiente. La variable A representa el conjunto de áreas, y el parámetro A_s simboliza el subconjunto de áreas donde se despliega el *slice s*.

Para cada vertical s y para cada área $a \in A_s$, se utiliza la fórmula de Shannon para estimar la tasa media experimentada en un instante de tiempo t en cada BS física b, $\rho_{s,a}(t)$, tal y como se muestra a continuación:

$$\rho_{s,a}(t) = W_{s,a}(t) \cdot \log_2\left(1 + \frac{P_{s,a}(t)}{\sigma^2 + I_{s,a}(t)}\right)$$
(3.1)

donde $W_{s,a}(t)$ se corresponde con la cantidad de recursos físicos de $b \in \mathcal{B}$ asignados al *slice* $s \in S$ en el área $a \in \mathcal{A}_s$ en el *slot* t. Las variables $P_{s,a}(t)$ e $I_{s,a}(t)$ se refieren a la señal y a la interferencia medias experimentadas por los dispositivos del *slice* en ese área, en una ranura t.

 \mathcal{F} se define como el conjunto de *functional splits* posibles configurables en las BSs, y la variable $f_s^b(t)$ se corresponde con el *functional split* seleccionado para la BS virtual utilizada por el *slice* s en la BS física b. Además, $\alpha_s^b(t)$ representa la cantidad de recursos inalámbricos otorgados por la BS física $b \in \mathcal{B}$ a la BS virtual del servicio $s \in \mathcal{S}$ en el *slot* t.

Las componentes de la Ecuación 3.1 pueden depender de variables de decisión, de variables aleatorias o de ambas. Al considerar una variable aleatoria genérica $\omega(t)$, se pueden expresar las variables de la Ecuación 3.1 como funciones de la siguiente manera:

- $W_{s,a}(t) = \hat{W}(\alpha_s^b(t))$
- $I_{s,a}(t) = \hat{I}(\omega(t), f_s^b(t))$
- $P_{s,a}(t) = \hat{S}(\omega(t)).$

Se asume que las BSs que cubren un mismo área se coordinan para mejorar la eficiencia espectral. Sin embargo, una coordinación efectiva requiere que las BSs utilicen el mismo número de recursos. Por lo tanto,

$$W_{s,a}(t) = \hat{W}(\alpha_s^b(t)) = 180 \cdot 10^3 \cdot \min(\{\alpha_s^b(t) \mid b \in \mathcal{B}_a\})$$
(3.2)

donde \mathcal{B}_a se corresponde con el subconjunto de BSs físicas que cubren el área $a \in \mathcal{A}_s$ y el valor de 180 KHz representa el tamaño frecuencial de cada *Resource Block (RB)* en LTE. Este último parámetro debe ser tenido en cuenta en el cálculo del *throughput*, pero a nivel de formulación no influye su inclusión.

En la Tabla 3.3 se recogen los símbolos y las definiciones de los parámetros más relevantes para la formulación del problema. Además, se dividen en cuatro grupos, en función de si se trata de conjuntos, de constantes, de variables o de variables de decisión.

Símbolo	Definición			
Conjuntos				
S	Conjunto de servicios/slices			
${\mathcal B}$	Conjunto de BSs físicas			
\mathcal{A}	Conjunto de áreas			
\mathcal{A}_s	Conjunto de áreas donde se despliega $s \in S$			
\mathcal{A}^b	Conjunto de áreas cubiertas por la BS $b \in \mathcal{B}$			
\mathcal{B}_a	Conjunto de BSs físicas para $a \in \mathcal{A}$			
${\mathcal F}$	Conjunto de functional splits posibles			
Constantes				
η_b	Recursos inalámbricos físicos de la BS $b \in \mathcal{B}$ (en <i>Physical Resource Block (PRB)s</i>)			
d_s	Demanda de <i>throughput</i> del servicio $s \in S$ (en Mbps)			
Variables				
$\rho_{s,a}(t)$	Throughput del servicio s en el área a en el slot t (en Mbps)			
$P_{s,a}(t)$	Potencia media de señal recibida de las conexiones relacionada con el servicio $s \in S$			
	en el área $a \in \mathcal{A}_s$ en el <i>slot</i> t			
$I_{s,a}(t)$	Potencia media de interferencia de las conexiones relacionada con el servicio $s \in S$			
	en el área $a \in \mathcal{A}_s$ en el <i>slot</i> t			
$W_{s,a}(t)$	Recursos asignados al servicio $s \in S$ en el área $a \in A_s$ en el <i>slot</i> t			
Variables de decisión				
$\alpha_s^b(t)$	Recursos inalámbricos otorgados por la BS física $b \in \mathcal{B}$ a la BS virtual del servicio			
	$s \in \mathcal{S}$ en el <i>slot</i> t			
$f_s^b(t)$	Opción de <i>split</i> de la BS virtual para el servicio $s \in S$ en la BS física $b \in \mathcal{B}$ en el slot t			

Tabla 3.3: Símbolos y definiciones

3.2.2. Formulación del problema

El objetivo del problema consiste en minimizar la cantidad de recursos asignados, garantizando que se satisfaga la demanda media a lo largo del tiempo. Se define la media temporal de la variable y(t) de la siguiente manera:

$$\overline{y} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\{y(t)\}$$
(3.3)

De forma similar, se definen las medias temporales de dos de las variables declaradas en la Tabla 3.3:

$$\overline{\alpha_s^b} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^T \mathbb{E}\{\alpha_s^b(t)\}$$
(3.4)

$$\overline{\rho_{s,a}} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\{\rho_{s,a}(t)\}$$
(3.5)

Para cumplir el objetivo de minimizar los recursos asignados, se presenta el siguiente problema de optimización:

$$\min_{f(t),\alpha(t)} \quad \sum_{s\in\mathcal{S},b\in\mathcal{B}} \overline{\alpha_s^b}$$
(3.6)

$$\overline{\rho_{s,a}} \ge d_s \qquad \forall a \in \mathcal{A}_s, \forall s \in \mathcal{S}$$
(3.8)

$$\sum_{s} \alpha_{s}^{b}(t) \leq \eta_{b} \quad \forall b \in \mathcal{B}, \forall t$$
(3.9)

$$f_s^b(t) \in \mathcal{F} \quad \forall s \in \mathcal{S}, \forall t$$
 (3.10)

Este problema se puede resolver haciendo uso del algoritmo *drift-plus-penalty*. Para ello, el primer paso consiste en reescribir el problema de forma más estándar:

$$\min_{f(t),\alpha(t)} \quad \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \overline{\alpha_s^b}$$
(3.11)

$$d_s - \overline{\rho_{s,a}} \le 0 \quad \forall a \in \mathcal{A}_s, \forall s \in \mathcal{S}$$
(3.13)

$$\sum_{s} \alpha_{s}^{b}(t) \le \eta_{b} \quad \forall b \in \mathcal{B}, \forall t$$
(3.14)

$$f_s^b(t) \in \mathcal{F} \quad \forall s \in \mathcal{S}, \forall t$$
 (3.15)

A continuación, se define $E_{s,a}(t) = d_s - \rho_{s,a}(t)$ y $\overline{E_{s,a}} = \lim_{T \to \infty} \frac{1}{T} \sum_{t \in T} \mathbb{E}\{d_s - E_{s,a}(t)\} = d_s - \overline{\rho_{s,a}}$. Después, se especifica una cola virtual de la siguiente manera:

$$Z_{s,a}(t+1) = \max\{Z_{s,a}(t) + E_{s,a}(t), 0\}$$
(3.16)

Para solucionar este problema, en cada *slot* se comprueba la cola y el estado del sistema, y se resuelve el siguiente problema:

$$\min_{f(t),\alpha(t)} \quad V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}(t) \cdot E_{s,a}(t)$$
(3.17)

$$\sum_{s} \alpha_s^b(t) \le \eta_b \quad \forall b \in \mathcal{B} \tag{3.19}$$

$$f_s^b(t) \in \mathcal{F} \quad \forall s \in \mathcal{S}$$
 (3.20)

Este problema se puede abordar de dos maneras distintas en función de lo que se considere que sea más importante conseguir. Así, se presentan los problemas *P1* y *P2*: en el primero solo se concede importancia a la minimización de la asignación de recursos asumiendo niveles de centralización fijos, y en el segundo se tiene en cuenta tanto la minimización de la asignación de recursos como la selección del *functional split*.

Para el caso del problema *P1*, sólo se selecciona la cantidad de recursos α_s^b , así que el problema se desarrolla de la siguiente manera:

$$\min_{\alpha(t)} V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}(t) \cdot E_{s,a}(t)$$
(3.21)

$$= V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}^b(t) \cdot (d_s - \rho_{s,a}(t))$$
(3.22)

$$= V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}^b(t) \cdot (d_s - W_{s,a}(t) \cdot E_{eff_{s,a}})$$
(3.23)

$$= \sum_{s \in \mathcal{S}} \left(V \cdot \sum_{b \in \mathcal{B}} \alpha_s^b(t) + \sum_{a \in \mathcal{A}_s} Z_{s,a}(t) \cdot d_s - \sum_{a \in \mathcal{A}_s} Z_{s,a}(t) W_{s,a}(t) E_{eff_{s,a}} \right)$$
(3.24)

En la Ecuación 3.24 se puede despreciar el producto de la cola virtual por la demanda de *throughput*, debido a que no depende de la decisión, con lo que resulta:

$$\min_{\alpha(t)} \sum_{s \in \mathcal{S}} \left(V \cdot \sum_{b \in \mathcal{B}} \alpha_s^b(t) - \sum_{a \in \mathcal{A}_s} Z_{s,a}(t) W_{s,a}(t) E_{eff_{s,a}} \right)$$
(3.25)

Además, debido a que se quiere minimizar la cantidad de recursos asignados, cualquier recurso que no contribuya al *throughput* no se asignará. Por lo tanto, $W_{s,a}(t) = \alpha_s^b(t) \ \forall b \in \mathcal{B}_a$. Adicionalmente, las BSs que no cubran un área para un servicio no darán recursos. De esta forma, el problema se reduce a:

$$\min_{\alpha(t)} \sum_{s \in \mathcal{S}} \left(V \cdot \sum_{a \in \mathcal{A}_s} \sum_{b \in \mathcal{B}_a} W_{s,a}(t) - \sum_{a \in \mathcal{A}_s} Z_{s,a}(t) E_{eff_{s,a}} W_{s,a}(t) \right)$$
(3.26)

$$= \sum_{s \in \mathcal{S}} \left(V \sum_{a \in \mathcal{A}_s} |\mathcal{B}_a| W_{s,a}(t) - \sum_{a \in \mathcal{A}_s} Z_{s,a}(t) E_{eff_{s,a}} W_{s,a}(t) \right)$$
(3.27)

$$= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} W_{s,a(t)} \left(V | \mathcal{B}_a| - Z_{s,a}(t) E_{eff_{s,a}} \right)$$
(3.28)

Que se puede reescribir como:

$$\min_{\alpha(t)} \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} K_{s,a} \cdot W_{s,a}(t)$$
(3.29)

$$\{\alpha_s^b(t) \mid b \in \mathcal{B}_a\} = W_{s,a}(t), \ \forall a \in \mathcal{A}_s, \forall s \in \mathcal{S}$$
(3.31)

$$\sum_{s} \alpha_{s}^{b}(t) \le \eta_{b} \quad \forall b \in \mathcal{B}$$
(3.32)

donde $K_{s,a}(t) = V|\mathcal{B}_a| - Z_{s,a}(t)E_{eff_{s,a}}.$

En el caso del problema *P2*, se da importancia tanto a la minimización de la asignación de recursos como a la selección del *functional split*. Cabe destacar que el *functional split* se define a nivel de BS virtual, por lo que afecta a todas las áreas de la misma. Además, la cooperación de las BSs virtuales sobre un área solapada se define por la decisión de *split* menos centralizada. El problema se plantea de la siguiente manera:

$$\min_{\alpha(t)} V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}(t) \cdot E_{s,a}(t)$$
(3.33)

$$= V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}^b(t) \cdot (d_s - \rho_{s,a}(t))$$
(3.34)

$$= V \cdot \sum_{s \in \mathcal{S}, b \in \mathcal{B}} \alpha_s^b(t) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}_s} Z_{s,a}^b(t) \cdot \left(d_s - W_{s,a}(t) \cdot \log_2 \left(1 + \frac{P_{s,a}(t)}{\sigma^2 + I_{s,a}(t)} \right) \right)$$
(3.35)

$$=\sum_{s\in\mathcal{S}}\left(V\cdot\sum_{b\in\mathcal{B}}\alpha_s^b(t)+\sum_{a\in\mathcal{A}_s}Z_{s,a}(t)\cdot d_s-\sum_{a\in\mathcal{A}_s}Z_{s,a}(t)W_{s,a}(t)\cdot \log_2\left(1+\frac{P_{s,a}(t)}{\sigma^2+I_{s,a}(t)}\right)\right)$$
(3.36)

En la Ecuación 3.36 se puede descartar el producto de la cola virtual por la demanda de *throughput*, debido a que no depende de la decisión:

$$\min_{\alpha(t)} \sum_{s \in \mathcal{S}} \left(V \cdot \sum_{b \in \mathcal{B}} \alpha_s^b(t) - \sum_{a \in \mathcal{A}_s} Z_{s,a}(t) W_{s,a}(t) \log_2 \left(1 + \frac{P_{s,a}(t)}{\sigma^2 + I_{s,a}(t)} \right) \right)$$
(3.37)

3.3. Escenarios de aplicación

En este apartado se describen los tres escenarios de aplicación utilizados para analizar el problema descrito. En primer lugar, se utiliza un escenario básico, mostrado en la Figura 3.2, donde se considera que existen dos BSs físicas, BS1 y BS2, que proporcionan cobertura a sus respectivas áreas, A1 y A2, así como a una tercera área solapada, A3.



Figura 3.2: Primer escenario de aplicación

En la Figura 3.3 se muestra el segundo escenario de aplicación que se ha analizado. En este caso, se consideran tres BSs físicas (BS1, BS2 y BS3), donde BS1 cubre un amplio área, que abarca las zonas de cobertura de las otras dos BSs. De esta manera, BS2 y BS3 cubren sus respectivas áreas y BS1 abarca las tres áreas implementadas. Por tanto, existen dos áreas, A2 y A3, que se encuentran cubiertas por dos BSs al mismo tiempo.



Figura 3.3: Segundo escenario de aplicación

El último escenario de aplicación considerado es más complejo, y supone una combinación de los dos anteriores. Como se observa en la Figura 3.4, está compuesto por seis BSs físicas, donde cada una de ellas cubre sus respectivas áreas, además de las que se encuentren solapadas. Este escenario se diferencia de los anteriores en términos del número de áreas cubiertas por más de una BS. Para este caso, se aprecia que las áreas A4, A5, A6, A8, A9 y A10 son cubiertas por dos BSs a la vez, mientras que el área A7 está cubierta por tres BSs al mismo tiempo.



Figura 3.4: Tercer escenario de aplicación

Capítulo 4

Desarrollo e implementación

En este capítulo se describe en detalle el entorno de simulación desarrollado, así como los algoritmos implementados para resolver el problema descrito en el capítulo anterior. Además, se explican los parámetros de configuración necesarios para el funcionamiento correcto del entorno y de los algoritmos generados.

4.1. Parámetros de configuración

Para que el entorno de simulación y los algoritmos desarrollados se comporten de manera adecuada, es necesario definir en un fichero los parámetros de configuración más relevantes. Se han implementado tres ficheros de configuración, uno para cada escenario de aplicación considerado. Además, los parámetros que se van a especificar son comunes a todos los escenarios, y están descritos en formato JSON.

El primer parámetro de configuración incluido en los ficheros es el nivel de *functional split*. Permite indicar un valor de *split* estático para el caso en el que solo se tiene como objetivo la minimización de los recursos asignados. Para la implementación se han considerado cuatro *split*s posibles: C-RAN (todas las funciones centralizadas), MAC/PHY, RLC/MAC y PDCP/RLC.

El siguiente parámetro es la variable V, encargada de asignar mayor o menor peso a la minimización de recursos, en función de si presenta un valor más alto o más bajo, respectivamente. Por otro lado, en el fichero de configuración, también se especifica el valor de la complejidad computacional a asignar a la CU, siendo un aspecto clave a la hora de decidir la cantidad de funciones que se pueden centralizar. Después, se indica el nombre de cada BS física existente en el escenario de aplicación considerado y el número de recursos que presenta cada una.

Posteriormente, se describen las zonas del escenario, considerando la existencia de una zona por cada área y cada servicio desplegado en ella. Para cada una de estas zonas se proporciona el nombre que se le asigna, el nombre de las BSs físicas que la cubren y los rangos de señal e interferencia, expresados en mW, entre los que puede encontrarse la señal y la interferencia percibidas por los usuarios. Además, se especifica también el *throughput* objetivo y el *throughput* mínimo de cada zona, en Mbps, así como a qué *slice* de los implementados pertenece.

Otro parámetro de configuración que es necesario proporcionar es el nombre del algoritmo a implementar. Se han desarrollado tres algoritmos distintos, que se explican en la Sección 4.2, por lo que en el fichero se debe indicar cuál se quiere utilizar. Por último, se proporciona el número de iteraciones que tiene que llevar a cabo el programa.

4.2. Algoritmos implementados

En esta sección, se describen los tres algoritmos que se han desarrollado, en C++. Los dos primeros están enfocados en la minimización de la asignación de recursos y el tercero tiene en cuenta tanto dicha minimización como la selección del nivel de *split* adecuado. Para la implementación de estos algoritmos, se ha definido una clase padre que contiene funciones comunes a todos ellos y una clase hija por cada algoritmo, donde se añaden variables miembro particulares de cada uno. Los métodos de la clase padre son heredados por cada una de las clases hija, como se muestra en la Figura 4.1.



Figura 4.1: Diagrama de clases de los algoritmos implementados

La clase padre define, además de su constructor y destructor, cuatro métodos a heredar por las clases hijas: *Init, Run, Decision y RecursosrestantesBSfisicas*. La primera función tiene como objetivo la inicialización de todas las variables necesarias para ejecutar los algoritmos. La segunda función contiene el código principal del algoritmo y es la encargada de llevar a cabo su funcionalidad concreta. La función *Decision* devuelve al entorno de simulación los resultados obtenidos de la ejecución del algoritmo, es decir, la decisión de los recursos asignados a las BSs virtuales y, en el caso del tercer algoritmo, el *split* seleccionado también. La cuarta función retorna los recursos restantes de las BSs físicas tras la asignación de recursos a las BSs virtuales.

4.2.1. GLPK

Los tres algoritmos se ejecutan haciendo uso del *solver* GNU Linear Programming Kit (GLPK). GLPK constituye una herramienta que permite la resolución de problemas de programación lineal mediante la especificación de dichos problemas en forma matricial. De esta manera, mediante el *solver* GLPK, se obtiene la asignación de recursos óptima para el problema.

En este caso, se ha definido un *solver* GLPK para cada uno de los escenarios de aplicación y para dos aproximaciones, implementando cada *solver* como una clase. La primera aproximación (AI) considera que los recursos asignados a las BSs virtuales de una misma BS física y para un mismo servicio en distintas áreas, tienen que ser iguales, mientras que la segunda aproximación (A2) permite la posibilidad de que no

sea así. Sin embargo, ambas alternativas coinciden en que se deben asignar el mismo número de recursos a las BSs virtuales de un mismo servicio en caso de que se encuentren en un área solapada.

Cada una de las clases GLPK definidas utiliza la librería GLPK, y está formada por tres métodos: *Init, Run y Decision.* La función *Init* realiza la inicialización de las variables miembro, recibiendo como parámetros los coeficientes de cada una de las zonas existentes, los recursos de las BSs físicas implementadas en el escenario de aplicación analizado y los recursos mínimos que se deben proporcionar en cada una de las zonas.

Con esta información, en la función *Run*, se crea una matriz en base a las restricciones definidas en las Ecuaciones (3.31) y (3.32). Las filas de dicha matriz se corresponden con las limitaciones de capacidad de las BSs y con las restricciones de las zonas, y las columnas especifican las variables que aparecen en dichas limitaciones y las incógnitas a resolver en el problema. Para el caso de la aproximación *A2*, se reducen las filas, indicando únicamente las restricciones de capacidad de las BSs y las limitaciones de las zonas solapadas. Una vez descrita la matriz, el *solver* GLPK resuelve el problema, y proporciona como resultados los recursos a asignar a las BSs virtuales, así como el valor óptimo conseguido.

La función *Decision* se encarga de retornar los recursos a asignar a las BSs virtuales a la clase del algoritmo correspondiente. Además, para el caso del tercer escenario de aplicación y para la implementación de la aproximación *A2*, se define en la clase GLPK un método adicional, denominado *Optimalvalue*, que devuelve el valor óptimo conseguido al entorno de simulación, siendo este parámetro fundamental en la decisión del *split* adecuado.

4.2.2. Primer algoritmo

Una vez explicado el funcionamiento del *solver* GLPK, se procede a describir cada uno de los algoritmos implementados. El primer algoritmo desarrollado se basa en la aproximación A1 y la Figura 4.2 ilustra su diagrama de flujo. En primer lugar, en este algoritmo se calculan los coeficientes de cada zona existente en el escenario de aplicación considerado, mediante la aplicación de la expresión $K_{s,a}(t) = V|\mathcal{B}_a| - Z_{s,a}(t)E_{eff_{s,a}}$. Después, se obtienen los recursos de las BSs físicas desplegadas, así como los recursos mínimos que deben ser garantizados en cada zona.

Con estos datos, se llama al *solver* GLPK correspondiente a la aproximación *A1* y al escenario de aplicación que se quiera evaluar. Este *solver* resuelve el problema de optimización y devuelve como resultado la cantidad de recursos óptima a asignar por cada zona. Se asignan dichos recursos a las BSs virtuales existentes en cada zona y se calculan los recursos restantes de las BSs físicas. Para ello, para cada BS física y para cada servicio, se obtiene el máximo número de recursos asignados de entre todas las BSs virtuales que pertenecen a ellos, calculando así los recursos restantes de las BSs físicas como la diferencia entre sus recursos originales y los máximos obtenidos por cada servicio.

4.2.3. Segundo algoritmo

El segundo algoritmo implementado presenta un esquema muy similar al anterior, por lo que el diagrama de flujo mostrado en la Figura 4.2 también es válido para este caso. Las funciones principales que lleva a cabo coinciden, pero introduce ciertas diferencias en su implementación. Este algoritmo se basa en la aproximación A2, a diferencia del anterior, que se enfocaba en la aproximación A1.

De esta manera, se calculan los coeficientes de cada una de las zonas del escenario de aplicación escogido y, junto con los recursos de las BSs físicas existentes y con los recursos mínimos que se deben garantizar en cada zona, se llama al *solver* GLPK correspondiente a la aproximación *A2* y al escenario utilizado. Este *solver* devuelve como resultado los recursos óptimos a asignar por cada BS física en cada zona. De esta forma, se asignan los recursos obtenidos a las BSs virtuales correspondientes, y se calculan los recursos restantes de las BSs físicas. En este caso, en vez de considerar el mayor número de recursos asignados, se restan a los recursos originales de cada BS física los recursos asignados a cada BS virtual que pertenezca a ella.



Figura 4.2: Diagrama de flujo de los dos primeros algoritmos

4.2.4. Tercer algoritmo

El tercer algoritmo, además de centrarse en la minimización de los recursos asignados a las BSs virtuales, también permite la selección del nivel de *split* adecuado, basándose en la aproximación *A2*. Este algoritmo, cuyo diagrama de flujo se muestra en la Figura 4.3, se utiliza para el tercer escenario de aplicación, considerando la elección del nivel de centralización para el segundo servicio en las áreas 8, 9 y 10.

En primer lugar, se generan todas las combinaciones existentes de nivel de *split* para esas áreas. Cada zona puede presentar un *functional split* comprendido entre el 1 y el 4, siendo el 1 la menor centralización posible (PDCP/RLC) y aumentando sucesivamente hasta llegar al 4, que presenta el mayor *split* (C-RAN). Por tanto, existen en total 64 combinaciones de *split*.

A continuación, se determinan qué combinaciones de las existentes cumplen el requisito de complejidad computacional. Para ello, se obtiene para cada zona la carga computacional asociada a su nivel de *split* y se comprueba si la suma de las cargas para las tres zonas no sobrepasa un valor de carga máximo establecido, es decir, que no satura la CU. Se realiza este proceso con todas las combinaciones existentes, y se seleccionan aquellas que son realizables, debido a que cumplen el requisito de complejidad computacional.

Posteriormente, para cada una de las combinaciones seleccionadas, se obtienen los coeficientes de cada una de las zonas existentes en el tercer escenario. Para calcular estos coeficientes, se necesita conocer el valor de la eficiencia espectral y, por tanto, de la SINR. Por ello, en las zonas donde se aplica el *functional split*, dependiendo de su valor se consigue un factor de cancelación de la interferencia concreto, como se muestra en la Tabla 4.1. Entonces, para el caso de las áreas 8, 9 y 10, se obtiene el factor de cancelación de la interferencia correspondiente a los niveles de *split* analizados en ese momento, y se aplica sobre la interferencia.

Por otro lado, haciendo uso de las SINRs obtenidas para estas zonas, tras implementar el factor de cancelación de la interferencia, se obtienen los recursos mínimos a garantizar en cada zona, mediante el uso de la expresión del *throughput*, definida en la Ecuación (3.1). El resto de zonas donde no se realiza una selección dinámica del *functional split*, utilizan los valores de SINR y de recursos mínimos calculados previamente en el entorno de simulación y tomados como datos de entrada del algoritmo.

Functional split	Factor de cancelación		
C-RAN	0.01		
MAC/PHY	0.2		
RLC/MAC	0.6		
PDCP/RLC	1		

Tabla 4.1: Factor de cancelación de la interferencia para cada functional split

Una vez conocidos los coeficientes de cada una de las zonas, los recursos de las BSs físicas y los recursos mínimos a garantizar en cada zona, para una combinación concreta de *split*s, se procede a llamar al *solver* GLPK correspondiente al tercer escenario de aplicación y a la aproximación A2. Este *solver* devuelve como resultado los recursos a asignar por cada BS física en cada zona y el valor óptimo conseguido. Tras realizar este proceso con todas las combinaciones de *split* posibles, se selecciona aquella que presente el menor valor óptimo.

Seguidamente, se asigna el nivel de *split* a las áreas 8, 9 y 10, de acuerdo a la combinación seleccionada. Además, se especifican sus SINRs y sus recursos mínimos asociados a dicha combinación. De esta forma, se asignan los recursos obtenidos del *solver* para la combinación definitiva a las BSs virtuales correspondientes, y se calculan los recursos restantes de las BSs físicas mediante la resta de los recursos originales de cada BS física y los recursos asignados a cada BS virtual que pertenezca a ella.



Figura 4.3: Diagrama de flujo del tercer algoritmo

4.3. Entorno de simulación

La plataforma desarrollada para la evaluación de los algoritmos descritos en la sección anterior se ha implementado en C++ y permite, a partir de un fichero de configuración, analizar el funcionamiento de dichos algoritmos y proporcionar los resultados oportunos. Para llevar a cabo este proceso, la plataforma implementa cinco clases fundamentales: *BSfisica*, *BSvirtual*, *Service*, *Zona* y *Maestra*. Por otro lado, una vez generada la plataforma, se realiza su automatización para obtener los ficheros de resultados de manera rápida y representarlos de forma gráfica.

4.3.1. Clases implementadas

La clase *BSfisica* representa a las BSs físicas, y posibilita la creación de una instancia de esta clase por cada una de las BSs físicas existentes en el escenario de aplicación considerado y, por tanto, descritas en el fichero de configuración. Cada una de las instancias de esta clase permite el almacenamiento de los recursos de cada BS física, de los servicios que posibilita, de los nombres y recursos de las BSs virtuales

pertenecientes a ella y del *slot* que se está analizando. Todos estos datos son guardados en las instancias, y se accede a ellos a través de los métodos definidos en la clase.

La clase *BSvirtual* representa a las BSs virtuales y permite generar una instancia por cada BS virtual considerada. Se asume que existe una BS virtual perteneciente a una BS física por cada área que cubra dicha BS física y por cada servicio desplegado en ese área. Los métodos de esta clase posibilitan la lectura y el almacenamiento de los recursos asignados a las BSs virtuales, del nivel de *split* y del servicio y de la BS física a la que pertenecen. Además, esta clase posibilita el reinicio (*reset*) de los recursos previamente asignados a las BSs virtuales.

La clase *Service* hace referencia a cada uno de los *slices* desplegados en cada área, por lo que por cada zona considerada se crea una instancia de la clase *Service*. Estas instancias guardan y facilitan el acceso a la demanda de cada servicio, a su cola virtual, al *slice* al que pertenecen, a su SINR, a la zona asociada a ellos y al *slot* analizado. Esta información, como en el resto de las clases, es accesible a través de los métodos de la misma.

La clase *Zona* representa a las zonas presentes en el escenario de aplicación analizado, considerando la existencia de una zona por cada área y cada servicio existente en ella. Por lo tanto, por cada zona definida se crea una instancia de la clase *Zona*. Cada una de ellas permite la lectura y el almacenamiento de las BSs físicas que dan cobertura en esa zona, de la demanda existente, de las BSs virtuales que pertenecen a ella, del *throughput* experimentado en dicha zona, de su SINR, de los recursos mínimos y del *throughput* mínimo que se debe garantizar, así como de la señal y la interferencia experimentadas en esa zona.

La clase *Maestra* es la encargada de ejecutar todas las funcionalidades de la plataforma y de comunicarse con cada una de las clases descritas anteriormente. Está formada por dos métodos principales: *Init*, donde se realiza la inicialización de todas las variables y la creación de las instancias necesarias, y *Run*, donde se lleva a cabo la evaluación de los algoritmos y donde se proporcionan los resultados obtenidos.

Esta clase recibe como parámetro el fichero de configuración correspondiente al escenario de aplicación que se quiere analizar. A partir de él, en la función *Init*, se toman los datos pertinentes y se almacenan en variables miembro de la clase. Además, se crean las instancias necesarias de cada una de las clases descritas previamente, acorde con lo indicado en el fichero de configuración. De esta manera, se obtiene el *functional split* estático fijado y, por cada BS física, se crea su instancia y se almacenan sus recursos. También, por cada una de las zonas, se generan sus instancias y se guardan las BSs físicas presentes en ellas. Se almacena el rango de señal, el rango de interferencia, el *throughput* objetivo, el *throughput* mínimo y el *slice* al que pertenece cada zona.

El método *Init* permite la creación de una instancia de un servicio por cada zona, guardando en ella el *slice* y la zona a las que pertenece. Además, facilita el almacenamiento de dichos servicios en las BSs físicas correspondientes. Posibilita la generación de instancias de las BSs virtuales oportunas, guardando en ellas los servicios, las BSs físicas y las zonas a las que pertenecen, así como el nivel de *split* estático asignado. Además, se almacenan los nombres de las BSs virtuales en las BSs físicas y en las zonas correspondientes.

Después, en función del nombre de algoritmo indicado en el fichero de configuración, se crea una instancia de la clase algoritmo con dicho nombre, siendo posible la selección de cualquiera de las tres alternativas detalladas en la sección anterior. Por otro lado, en este fichero se indica también el número de

iteraciones que se deben llevar a cabo en el programa, trasladando su valor a una variable miembro para su uso posterior.

Una vez detalladas las instancias creadas y las variables miembro inicializadas, se procede a la ejecución de la función *Run*. En primer lugar, se inicializan las colas virtuales de las zonas definidas, se determinan el número y los nombres de los *slices* considerados y se crean los ficheros de resultados. Después, se obtiene el *throughput* objetivo de cada uno de los servicios, y se almacena en las instancias de los servicios adecuados. También, para cada una de las zonas, se asigna el valor de demanda de los servicios correspondiente. A continuación, se lleva a cabo la evaluación del algoritmo seleccionado, el número de veces especificado en la variable correspondiente al número de iteraciones, definidas en el fichero de configuración.

Para cada iteración, se realiza un *reset* de los recursos asignados a las BSs virtuales en el *slot* anterior. Además, se calcula la SINR de cada una de las zonas consideradas. Para ello, se obtienen los rangos de señal y de interferencia de cada una de las zonas, definidos en el fichero de configuración. A partir de ellos, se genera un valor aleatorio de señal y otro de interferencia comprendidos en dichos rangos y se guardan en la instancia de la zona adecuada. En caso de que exista más de una BS física cubriendo la zona, se obtiene el nivel de centralización de las BSs virtuales existentes en ella y se aplica a la interferencia el factor de cancelación oportuno. Considerando σ^2 como 10^{-3} W/Hz, se obtiene la SINR de cada zona como $SINR = \frac{P_{s,a}(t)}{\sigma^2 + I_{s,a}(t)}$. Cada valor de SINR obtenido se almacena en la instancia de la zona y del servicio correspondientes.

Después, se calculan los recursos mínimos a garantizar en cada una de las zonas. Para ello, a partir de la SINR y del *throughput* mínimo de cada zona, se llama a una función que obtiene los recursos mínimos, haciendo uso de la expresión del *throughput* definida en la Ecuación (3.1). Estos recursos se almacenan en la instancia de la zona asociada. Una vez generados todos los parámetros necesarios, se inicializa y ejecuta el algoritmo seleccionado, devolviendo como resultado tanto la asignación de recursos a las BSs virtuales como los recursos restantes de las BSs físicas. En caso de utilizar el tercer algoritmo, éste retorna también la decisión del nivel de *split* llevada a cabo.

El siguiente paso consiste en el almacenamiento de los recursos asignados a las BSs virtuales en el *slot* analizado en las BSs físicas, con el objetivo de poder devolver estos datos como resultados de la simulación posteriormente. A continuación, se calcula el *throughput* experimentado en cada una de las zonas. Para ello, se determina el número de recursos asignados a las BSs virtuales desplegadas en cada zona y, junto con la SINR de cada una, se aplica la expresión del *throughput*, definida en la Ecuación (3.1). Se almacena dicho *throughput* en la instancia de la zona correspondiente. Seguidamente, se realiza el cálculo de la cola virtual de cada zona, siguiendo la expresión de la Ecuación (3.16). De esta manera, se obtiene el valor de la cola en el siguiente *slot* mediante la selección del valor más alto entre 0 y el valor de la cola en el *slot* anterior junto a la diferencia entre la demanda existente en esa zona y el *throughput* proporcionado en ella. La cola virtual obtenida por cada zona se guarda en la instancia del servicio correspondiente.

Tras la realización de este proceso el número de veces indicado en el fichero de configuración, se procede a presentar los resultados obtenidos en diversos ficheros. Se crean seis tipos de ficheros relativos a: (1) los recursos asignados, (2) el *split* seleccionado, (3) las colas virtuales, (4) los recursos restantes de las BSs físicas, (5) la SINR y (6) el *throughput*. La primera clase de fichero recoge, en función del *slot*,

los recursos asignados a las BSs virtuales de una BS física concreta, por lo que se crea un fichero de este tipo por cada BS física existente en el escenario de aplicación.

El segundo fichero incluye, en función del *slot*, el nivel de *split* asignado a cada una de las BSs virtuales pertenecientes a una BS física, por lo que se genera un fichero de esta clase por cada BS física presente en el escenario. En el tercer fichero se almacena la cola virtual de cada una de las zonas definidas en función del *slot*. El cuarto fichero recoge los recursos restantes de cada una de las BS físicas existentes en función del *slot*. El quinto fichero muestra, en relación al *slot*, la SINR experimentada en cada una de las zonas definidas, mientras que el sexto fichero almacena los valores de *throughput*, en Mbps, conseguidos en cada zona y en todos los *slots* del experimento.

En la Figura 4.4 se observa el diagrama de flujo de la clase *Maestra*. Por otro lado, en la función principal del programa (*main*), se instancia esta clase, y se llama a sus dos métodos para realizar la ejecución de todo el experimento.



Figura 4.4: Diagrama de flujo de la clase Maestra

4.3.2. Automatización de la plataforma y herramientas utilizadas

Para automatizar la ejecución de la plataforma, se crea un fichero en Python que permite variar ciertos parámetros de configuración y almacenar los resultados obtenidos. En primer lugar, en este archivo se realiza la copia del fichero de configuración del escenario de aplicación considerado en un fichero auxiliar y se determina la ruta (*path*) donde se guardan los resultados generados. Después, se escoge qué parámetro de configuración se desea variar, teniendo las siguientes opciones: el *throughput* mínimo, el *throughput* objetivo del segundo *slice*, la variable V o la capacidad computacional.

El número de ejecuciones del programa depende del parámetro a variar y del rango de valores que tome. Para cada iteración, se obtienen los datos del fichero de configuración auxiliar y se especifica en dicho fichero el nivel de *split* estático que se quiere fijar. Además, se almacenan los identificadores de cada una de las BSs físicas indicadas en el fichero, con el objetivo de poder usarlos a la hora de crear los ficheros de resultados. En caso de que se varíe la complejidad computacional, se escribe el valor asignado en cada iteración en la variable correspondiente del fichero auxiliar.

Si el parámetro modificado se corresponde con el *throughput* objetivo del segundo *slice*, para cada una de las zonas donde se despliegue dicho *slice*, se realiza el aumento del valor del *throughput* objetivo en cada iteración en función del rango que se indique y se mantiene a un valor fijo el *throughput* objetivo del primer *slice*. También se puede alterar el valor del *throughput* mínimo de la zona solapada del primer escenario de aplicación y, en concreto, para el segundo *slice*. Para ello, en dicha zona, se obtiene su *throughput* objetivo y se establece el valor mínimo como un porcentaje determinado de aquel. De esta manera, con cada iteración realizada se aumenta dicho porcentaje y, por tanto, el *throughput* mínimo de la zona solapada.

El último parámetro que se puede modificar es la variable V. Se establece un rango de valores que puede tomar y, en cada iteración, se le asigna uno de ellos en orden ascendente. Una vez se ha realizado la modificación del parámetro elegido, se actualiza el fichero de configuración auxiliar con el nuevo valor. A continuación, se procede a compilar y ejecutar el programa, pasándole como argumento el fichero auxiliar actualizado. De esta forma, se llevan a cabo todas las funcionalidades del programa descrito anteriormente, evaluando el algoritmo correspondiente y generando los ficheros de resultados para cada iteración especificada.

En cada una de estas iteraciones, se realiza una copia de los ficheros de resultados creados tras la ejecución del programa, guardándolos en el *path* especificado al principio del fichero de automatización. Los ficheros correspondientes al nivel de *split* y a los recursos asignados a las BSs virtuales de una BS física se almacenan indicando en el nombre de cada fichero el identificador de la BS física a la que hacen referencia. Por lo tanto, en cada iteración, se realiza la copia y el almacenamiento de los ficheros referentes a la cola virtual, a los recursos restantes de las BSs físicas, a la SINR, al *throughput*, y al nivel de *split* y a los recursos asignados a las BSs virtuales por cada BS física.

A partir de los ficheros de resultados generados en cada iteración, se obtienen una serie de gráficas que presentan los resultados más relevantes sobre el funcionamiento de los algoritmos analizados. El entorno utilizado para crear estas gráficas es *Jupyter Notebook*, una herramienta que permite ejecutar diversos fragmentos (*snippet*) de código paso a paso y de manera independiente, posibilitando la observación de los resultados obtenidos al instante. Además, facilita la creación, la modificación y la visualización de las

gráficas de resultados.

Estas gráficas han sido generadas a partir de las herramientas proporcionadas por las librerías *Matplotlib* y *Pandas*. La primera librería permite la creación de las gráficas necesarias y la representación en las mismas de los datos adecuados. Favorece la generación de diversas gráficas dentro de una misma figura y la especificación de los nombres de los ejes y de los títulos de las mismas. También permite la creación de leyendas y el almacenamiento de las figuras creadas. La segunda librería posibilita el manejo de la información de los ficheros de resultados de forma más sencilla, mediante el uso de *dataframes*, y facilita además la representación de los datos incluidos en ellos.

El desarrollo de la plataforma, incluyendo la creación de los ficheros de configuración, la implementación de los algoritmos y la generación de las gráficas de resultados, se ha llevado a cabo en el editor de código *Visual Studio Code*. Este editor posibilita el desarrollo del código necesario para el funcionamiento de la plataforma, tanto en C++ como Python, y facilita la creación de gráficas en este mismo entorno. Tras la descripción de todos los componentes de la plataforma: los ficheros de configuración, los algoritmos y el entorno de simulación, se muestra en la Figura 4.5 un esquema de la interacción existente entre ellos.



Figura 4.5: Esquema de la plataforma implementada

Como se observa en la Figura 4.5, en primer lugar se ejecuta el fichero de automatización, desarrollado en *Python*, el cual permite la variación de los valores de los parámetros de configuración comentados anteriormente, escribiendo los cambios realizados en un fichero auxiliar. Después, se proporciona el fichero de configuración modificado como parámetro de entrada a la clase *Maestra*, siendo ésta la encargada de llamar al algoritmo seleccionado para su evaluación. Este proceso se lleva a cabo un número determinado de veces; en concreto, en los experimentos llevados a cabo en este trabajo se realizan 10000 ejecuciones independientes. Posteriormente, una vez terminadas, se generan los ficheros de resultados apropiados, que se devuelve al fichero de automatización para su almacenamiento. La ejecución del programa completo

también se hace varias veces, dependiendo su número del parámetro de configuración modificado y de los valores concretos que tome. Una vez terminadas tanto las 10000 ejecuciones de la clase *Maestra* como las iteraciones correspondientes del fichero de automatización, se crean las gráficas de resultados adecuadas en *Jupyter Notebook*.

Capítulo 5

Resultados

En este capítulo se presentan los resultados obtenidos a partir de la evaluación de los diferentes algoritmos implementados, teniendo en cuenta los diversos escenarios de aplicación considerados y las dos aproximaciones tomadas.

5.1. Validación del algoritmo

Esta primera sección recoge los resultados obtenidos relacionados con la validación del correcto funcionamiento del algoritmo. Se analiza el comportamiento de los dos algoritmos iniciales para el primer escenario de aplicación en términos de colas virtuales y de *throughput*. Concretamente, se estudia la evolución de las colas virtuales para todas las zonas, y se observa el impacto de variar el *throughput* mínimo y la variable V en la cola virtual y el *throughput* experimentado en la zona 6 del primer escenario. Para obtener estos resultados, es necesario determinar los valores de los parámetros de configuración que permiten realizar la ejecución de la plataforma, recogidos en la Tabla 5.1. Estos parámetros son comunes a todos los escenarios evaluados, y sus valores se modifican según sea necesario, utilizando en cada caso aquellos que sean adecuados en función del algoritmo analizado.

Configuración				
Número de PRBs	100			
Interferencia (mW)	[1, 5]			
Señal (mW)	[1, 81]			
Nivel de split	1			
Parámetro V	5000			
Complejidad computacional CU	18			
Número de iteraciones	10000			
Slices				
Número de slices	2			
Throughput mínimo 1 (Mbps)	1			
Throughput objetivo 1 (Mbps)	10			
Throughput mínimo 2 (Mbps)	1			
Throughput objetivo 2 (Mbps)	10			

Tabla 5.1:	Parámetros	de	configura	ación
------------	------------	----	-----------	-------

5.1.1. Colas virtuales de las zonas del primer escenario

En primer lugar, se evalúa el primer algoritmo para el escenario de aplicación inicial y para la aproximación AI. Para este caso, se estima que todas las BSs físicas existentes en el escenario presentan 100 recursos, y que las BSs virtuales creadas cuentan con un nivel de *split* fijo de 1. Los rangos de señal y de interferencia definidos, así como el valor del parámetro V y del número de iteraciones son los mostrados en la Tabla 5.1.

En cuanto a la complejidad computacional, al utilizar el primer algoritmo, no se tiene en cuenta dicho parámetro, debido a que no se asignan niveles de *split* dinámicos. También se considera el despliegue de dos *slices*, siendo su *throughput* mínimo de 1 Mbps y su *throughput* objetivo de 10 Mbps, aunque esta última variable aumenta de forma progresiva, para el segundo *slice*, hasta 80 Mbps.

Una vez definidos los parámetros de configuración asociados al primer escenario y al primer algoritmo, se procede a evaluar el funcionamiento de este último. Para ello, como se observa en la Figura 5.1, se representa la evolución de las colas virtuales vinculadas a las zonas existentes del escenario para dos casos particulares. En concreto, para ambos casos, se muestra la acumulación de cada una de las colas hasta cierto instante, dividida entre el *slot* actual.

Las colas virtuales están asociadas a diversas zonas o, lo que es lo mismo, a diversos servicios desplegados en ellas. Los tres primeros servicios se corresponden con el primer *slice* y los tres últimos con el segundo *slice*. La Figura 5.1a hace referencia a tener un *throughput* objetivo del segundo *slice* de 10 Mbps y la Figura 5.1b a presentar un *throughput* objetivo para dicho *slice* de 60 Mbps.





(b) Throughput objetivo del slice 2 de 60 Mbps



Como se observa en la Figura 5.1a, las colas virtuales de las seis zonas existentes en el primer escenario se estabilizan. Por lo tanto, todas las zonas consiguen cumplir el requisito de *throughput* objetivo (10 Mbps) a lo largo del tiempo. Sin embargo, para la Figura 5.1b se aprecia la inestabilidad de las colas virtuales para todas las zonas. Esto se debe a que el *throughput* objetivo del segundo *slice* es de 60 Mbps, valor que no se puede cumplir, ya que el objetivo del primer algoritmo es minimizar los recursos asignados, a la vez que se garantiza un reparto equitativo de recursos para cada BS física.

Por tanto, este algoritmo intenta cumplir el throughput objetivo de ambos slices, pero no es capaz

de lograrlo, debido a su configuración. Además, se observa que los servicios pertenecientes al primer *slice* presentan menores valores de cola virtual que los correspondientes al segundo *slice*. La explicación reside en que el *throughput* objetivo del segundo *slice* es superior y, por lo tanto, la diferencia entre el *throughput* conseguido en cada momento y el objetivo es mayor.

De esta manera, se ha comprobado que el primer algoritmo posibilita la estabilización de las colas virtuales para un *throughput* objetivo del segundo *slice* de 10 Mbps, y no alcanza la estabilidad al aumentar los requisitos más allá de la capacidad del sistema, *throughput* objetivo de 60 Mbps. Así, se demuestra que este algoritmo presenta un comportamiento adecuado y coherente.

A continuación, se evalúa el funcionamiento del segundo algoritmo para el primer escenario de aplicación, teniendo en cuenta la aproximación A2. Los valores de los parámetros de configuración son los mismos que los del caso anterior. La Figura 5.2 muestra la evolución de las colas virtuales de las seis zonas para el segundo algoritmo, pero para dos valores distintos de *throughput* objetivo del segundo *slice*: 10 Mbps y 30 Mbps.





En la Figura 5.2a se aprecia que las colas virtuales de las zonas del escenario analizado se estabilizan, es decir, que cumplen el requisito de *throughput* objetivo a lo largo del tiempo. De hecho, en comparación con el caso anterior, estas colas se encuentran estabilizadas de manera más uniforme y a menor valor. Con respecto a la Figura 5.2b, se observa que si el *throughput* objetivo del segundo *slice* es de 30 Mbps, las colas son inestables, pero presentando valores más bajos que los proporcionados por el primer algoritmo.

Sin embargo, para las áreas no solapadas de ambos *slices*, la inestabilidad no es tan pronunciada, debido a que el segundo algoritmo permite asignar recursos de manera independiente entre BSs virtuales de una misma BS física, siempre con el objetivo de minimizar la asignación. En el área solapada, para ambos *slices*, al requerir una asignación igualitaria de recursos, los valores de las colas crecen, aunque sigue siendo superior la del segundo *slice* ya que presenta un *throughput* objetivo mayor. De esta forma, se comprueba que el segundo algoritmo también tiene un comportamiento adecuado y acorde con lo esperado. Además se observa que alcanza mayor estabilidad al tener un menor número de restricciones.

5.1.2. Throughput y cola virtual de la zona 6 del primer escenario

A continuación, con el objetivo de seguir evaluando el correcto funcionamiento de ambos algoritmos, se analiza el impacto de variar el *throughput* mínimo y la variable V en el área 3 del primer escenario de aplicación para el segundo *slice* o, lo que es lo mismo, en la zona 6. Para ello, se estudia la evolución del *throughput* y de la cola virtual en dicha zona, en función de los valores tomados tanto del *throughput* mínimo como del parámetro V. En concreto, el *throughput* mínimo presenta valores entre 1 Mbps y 3 Mbps, con un incremento de 0.5 Mbps. La variable V adquiere valores comprendidos entre 1000 y 9000, con pasos de 2000.

En primer lugar, se evalúa el primer algoritmo para el primer escenario de aplicación y para la aproximación A1. Los parámetros de configuración son los mismos que en los casos anteriores, con la excepción de que el *throughput* mínimo y la variable V ya no son fijos, sino que presentan los rangos mencionados anteriormente. Además, se fija el *throughput* objetivo del segundo *slice* a 10 Mbps. Con estos datos, tras la ejecución del algoritmo, se obtienen los resultados mostrados en la Figura 5.3, donde se representa el *throughput* y la cola virtual de la zona 6 del escenario para diversos valores del *throughput* mínimo y del parámetro V.





(a) *Throughput* de la zona 6 en función del *throughput* mínimo

(b) Throughput de la zona 6 en función de V



(c) Cola virtual de la zona 6 en función del *throughput* mínimo

(d) Cola virtual de la zona 6 en función de V

Figura 5.3: *Throughput* y cola virtual de la zona 6 en función del *throughput* mínimo y de V para el primer algoritmo

En la Figura 5.3a se representa el *throughput* experimentado en la zona 6 del primer escenario en función del valor del *throughput* mínimo existente en dicha zona utilizando un diagrama de caja o *box plot*. Este elemento permite visualizar la distribución de los datos obtenidos, especificando la siguiente información: la mediana, el primer y tercer cuartil, el rango entre cuartiles, los límites superior e inferior, y los valores atípicos. La mediana se representa con una línea horizontal dentro la caja.

El primer cuartil indica el valor por debajo del cual se encuentran el 25 % de los casos (línea inferior de la caja) y el tercer cuartil aquel valor por debajo del cual se localizan el 75 % de los casos (línea superior de la caja). El rango entre cuartiles se define como la diferencia entre el tercer y el primer cuartil, correspondiéndose con el 50 % de los datos, representándose con la altura de la caja. Los límites superior e inferior son los valores máximo y mínimo de los datos respectivamente, separándose de la caja como mucho 1.5 veces el rango entre cuartiles. Los valores atípicos son aquellos que se encuentran por encima del límite superior o por debajo del límite inferior, que no se han incluido en este caso, para favorecer la representación. Adicionalmente, se añade el valor medio del *throughput* de la zona 6 como un círculo azul.

En la Figura 5.3a se observa que, a medida que se aumenta el *throughput* mínimo de la zona 6, también lo hace el *throughput* experimentado en dicha zona. Además, se aprecia que el límite inferior del *box plot* para cada *throughput* mínimo coincide con dicho valor, como era de esperar, ya que se corresponde con el valor mínimo que se debe proporcionar en dicha zona. Se destaca asimismo que el *throughput* medio de la zona 6 va disminuyendo para cada *throughput* mínimo considerado. Por lo tanto, se comprueba que en dicha zona se cumple el *throughput* objetivo establecido de 10 Mbps para ciertos casos únicamente.

La Figura 5.3b muestra la evolución del *throughput* de la zona 6 en función del parámetro V. Se observa que el *throughput* se mantiene constante al aumentar el valor de V. Sin embargo, el *throughput* medio existente en esa zona se va reduciendo al incrementar la variable V. Esto se debe a que el parámetro V representa la importancia dada a la minimización de la asignación de recursos con respecto a la reducción de la cola virtual, siendo más alta cuanto mayor sea su valor. Por lo tanto, para valores altos de V, no se cumple el *throughput* objetivo de 10 Mbps en la zona. Se puede concluir, por tanto, que encontrar un valor apropiado de V que sea alto y permita cumplir el requisito de *throughput* objetivo es complicado, y constituye uno de los principales retos de este tipo de problemas.

La Figura 5.3c permite visualizar el comportamiento de la cola virtual de la zona 6 en función del *slot* para diversos casos de *throughput* mínimo establecidos. Se aprecia que, a medida que aumenta el *throughput* mínimo, la cola presenta valores más altos y se vuelve inestable. La explicación reside en que si el *throughput* mínimo se incrementa, es necesario asignar más recursos a esa zona para garantizar ese *throughput*, por lo que quedan menos recursos disponibles para proveer el *throughput* objetivo establecido. Por ello, al ser más complicado cumplir este requisito, la cola virtual se vuelve inestable.

En la Figura 5.3d se representa la evolución de la cola virtual de la zona 6 en función del *slot* para diferentes valores de la variable V. Se observa que, a medida que se incrementa dicho parámetro, la cola virtual presenta un valor mayor, manteniendo más o menos su estabilidad. Esto se debe a que cuanto más alta es la variable V, mayor es la importancia que se da a la minimización de la asignación de recursos y, por tanto, es menos relevante la reducción de la cola virtual, por lo que toma valores más altos. Además, se destaca que la cola virtual, para la mayoría de los valores de V, es estable, debido a que en esos casos se cumple el requisito de *throughput* objetivo.

El análisis de la Figura 5.3 permite concluir que el primer algoritmo, para el caso de la zona 6, presenta unos resultados de *throughput* y de cola virtual adecuados y conformes a lo esperado, tanto en términos de variación del *throughput* mínimo de la zona, como de la modificación de la variable V. A continuación, en la Figura 5.4, se presentan estos mismos resultados obtenidos para el segundo algoritmo, utilizando el primer escenario de aplicación y la aproximación A2. Los parámetros de configuración coinciden con los usados anteriormente, manteniendo los rangos de valores del *throughput* mínimo y de la variable V.





(a) *Throughput* de la zona 6 en función del *throughput* mínimo



(c) Cola virtual de la zona 6 en función del *throughput* mínimo

(b) Throughput de la zona 6 en función de V



(d) Cola virtual de la zona 6 en función de V

Figura 5.4: *Throughput* y cola virtual de la zona 6 en función del *throughput* mínimo y de V para el segundo algoritmo

En la Figura 5.4a, se muestra la evolución del *throughput* experimentado en la zona 6 en función del *throughput* mínimo. Presenta un comportamiento muy parecido al del primer algoritmo, con la diferencia de que en esta gráfica los valores de *throughput* están más acotados dentro del *box plot* y son algo menores. Esto se debe a que el segundo algoritmo implementa la aproximación *A2*, que considera que los recursos asignados a las BSs virtuales de una BS física, para un mismo servicio y en distintas áreas, no tienen por qué ser los mismos, por lo que solo se asignan a la zona 6 los recursos que son necesarios, teniendo todas las BSs virtuales existentes en esa zona los mismos recursos, ya que se trata de una zona solapada. De esta forma, se obtienen valores más ajustados de *throughput*, no como en el caso del primer algoritmo que sí asigna el mismo número de recursos en las condiciones descritas, por lo que presenta valores más altos

de *throughput*. Además, el *throughput* medio para todos los casos de *throughput* mínimo es de 10 Mbps, cumpliendo el requisito de *throughput* objetivo para el segundo algoritmo.

La Figura 5.4b representa el *throughput* existente en la zona 6 en función del valor de la variable V. Como ocurría con el primer algoritmo, en este caso el *throughput* también se mantiene constante al aumentar la V. Sin embargo, adquiere valores más pequeños y acotados que en ese caso, debido a que el segundo algoritmo implementa la aproximación A2. Con respecto al *throughput* medio, también disminuye al incrementar la V pero, en este caso, es menor, siendo más cercanos sus valores al requisito de *throughput* objetivo de 10 Mbps.

En la Figura 5.4c se observa la evolución temporal de la cola virtual de la zona 6 para diferentes valores del *throughput* mínimo. Se aprecia que las colas son estables para todos los casos de *throughput* mínimo, presentando el mismo valor, a diferencia de lo que ocurría en el primer algoritmo. Además, dicho valor es menor que el que se obtuvo en el análisis previo. Todo esto se debe a que el primer algoritmo implementa la aproximación *A1*, que limita la capacidad de decisión debido a que presenta más restricciones en cuanto a la asignación de recursos. En cambio, el segundo algoritmo, al utilizar la aproximación *A2*, cuenta con mayor libertad a la hora de asignar los recursos, permitiendo la estabilización de las colas virtuales.

La Figura 5.4d permite visualizar la evolución de la cola virtual de la zona 6 para diversos valores del parámetro V. El comportamiento observado es muy parecido al del primer algoritmo, con la diferencia de que las colas virtuales ahora presentan ocupaciones más bajas. Además, en este caso, las colas virtuales para valores altos de V son más estables, debido a que su *throughput* medio asociado es más próximo a 10 Mbps y, por tanto, se encuentran más cerca de cumplir el requisito de *throughput* objetivo en comparación con el primer algoritmo.

De esta manera, se determina que el comportamiento del segundo algoritmo también es adecuado y conforme a lo esperado. A la vista de todos los resultados obtenidos, se decide proseguir con la generación de resultados con la aproximación A2, es decir, utilizando el segundo algoritmo para la minimización de la asignación de recursos, y el tercer algoritmo, ya que incorpora la selección dinámica de *split*. Se selecciona el segundo algoritmo debido a que presenta un comportamiento mejor que el primero, al consiguir un *throughput* medio más cercano al *throughput* objetivo y una mayor estabilidad en la cola virtual de la zona 6. Por otro lado, para el resto de simulaciones realizadas, basándose en los resultados del segundo algoritmo, se establece el valor de V a 5000 y el *throughput* mínimo de las zonas a 1 Mbps, debido a que se trata de valores adecuados que permiten cumplir los objetivos establecidos.

5.2. Análisis de los tres escenarios de aplicación

En esta sección se presentan los resultados obtenidos con el segundo algoritmo, aplicando la aproximación A2, para todos los escenarios de aplicación. En concreto, se analizan los recursos y el *throughput* medios obtenidos para cada nivel de centralización estático, y para todos los escenarios considerados. Los recursos medios se presentan para cada uno de los *slices* existentes. Además, se estudian dichos parámetros de forma más detallada para el escenario 3, comparando dos situaciones de interferencia, normal y alta. También se analiza el impacto de limitar la presencia del segundo *slice* a las áreas 8, 9 y 10 para el tercer escenario. Por último, se presentan los resultados asociados a la normalización de los recursos medios asignados en el escenario 3 para diversos niveles de *split* estáticos, considerando la existencia del segundo slice en ciertas zonas y una interferencia alta.

5.2.1. Recursos y throughput medios para los tres escenarios

En las Figuras 5.5 y 5.6, se muestran los resultados asociados a los recursos medios asignados a cada *slice* y al *throughput* medio experimentado para todos los escenarios de aplicación y para cuatro niveles de centralización estáticos. Además, también se representan estos parámetros para el tercer escenario de aplicación, con dos valores de interferencia, considerando el segundo *slice* bien en todas las zonas o solo en las áreas 8, 9 y 10. Los parámetros de configuración utilizados son los recogidos en la Tabla 5.1, teniendo en cuenta que los valores de la interferencia y del nivel de *split* van a variar en función de los resultados obtenidos. La complejidad computacional de la CU no se tiene en cuenta, debido a que los niveles de centralización utilizados son estáticos.



(a) Recursos medios asignados a cada *slice* para diferentes niveles de centralización y para distintos escenarios



(b) Recursos medios asignados a cada *slice* para diferentes niveles de centralización para el tercer escenario y para dos rangos de interferencia

Figura 5.5: Recursos medios asignados a cada *slice* en función de los niveles de centralización y de los escenarios

La Figura 5.5a muestra los recursos medios asignados a cada *slice* para los cuatro niveles de centralización estáticos considerados, y para los tres escenarios de aplicación. Los recursos medios se computan entre todas las BSs físicas existentes en cada escenario. Se observa que, a medida que se aumenta el nivel de *split*, se reduce el número de recursos necesarios para cada *slice*, debido a que se centralizan más funciones en la CU. Para el caso del tercer escenario, esto no ocurre debido a que la SINR de sus zonas alcanza el valor máximo establecido, 31, por lo que aunque se aumente el nivel de centralización, no hay ninguna consecuencia de la disminución de recursos, ya que se ha alcanzado el valor máximo de SINR. Este valor viene determinado por la eficiencia espectral de 4G en el DL, que es de 5 b/s/Hz. Comparando los tres escenarios de aplicación, se comprueba que el tercero realiza una mayor asignación de recursos en media, debido a que presenta un mayor número de zonas a las que dar cobertura. Además, el segundo escenario reparte un menor número de recursos que el primero ya que existen BS físicas en las zonas solapadas, que reparten menos recursos en media, debido a que solo deben cubrir dichas zonas.

La Figura 5.5b representa los recursos medios asignados entre todas las BS físicas a cada *slice* para los cuatro niveles de centralización estáticos y para el escenario 3. En concreto, estos resultados se muestran para el caso en el que existe un rango de interferencia normal, entre 1 mW y 5 mW, y para el supuesto en el que se sube el valor de dicha interferencia, entre 11 mW y 15 mW. Se observa que, al aumentar el valor de interferencia, el valor de la SINR de las zonas del escenario disminuye y ya no alcanza su valor máximo, por lo que ahora sí se aprecia el efecto de aumentar los niveles de centralización, resultando en la reducción de los recursos asignados en media a cada *slice*.

La Figura 5.6a permite visualizar la distribución del *throughput* medio observador en todas las zonas de cada escenario para los cuatro niveles de centralización definidos, y para los tres escenarios de aplicación considerados. Las cajas de color azul se corresponden con el primer escenario, las de color morado con el segundo y las de color cian con el tercero. Se observa que el *throughput* medio, representado por un círculo azul, presenta un valor de 10 Mbps para el primer y segundo escenario, comprobándose así que ambos cumplen el requisito de *throughput* objetivo. Sin embargo, en el tercer escenario, no se alcanza dicho valor para ninguno de los niveles de centralización, lo que significa que no se consigue cumplir con el rendimiento deseado. Por otro lado, el primer escenario presenta mayores valores de *throughput* medio que el segundo, ya que que realiza una asignación de recursos mayor. El tercer escenario cuenta con una distribución de *throughput* medio menor, debido a que no puede asignar el número necesario de recursos para cumplir el requisito de *throughput* objetivo, puesto que los recursos disponibles están limitados y las BSs físicas deben dar cobertura a muchas zonas. Por otro lado, para los tres escenarios, se aprecia el efecto de los niveles de centralización, ya que al aumentar su valor, también lo hace el *throughput* experimentado.

En la Figura 5.6b se representa la distribución del *throughput* medio existente entre todas las zonas del tercer escenario, para los cuatro niveles de centralización establecidos y para cuatro casos diferentes. El primero, simbolizado por las cajas de color cian, muestra el *throughput* medio teniendo en cuenta valores de interferencia normales, es decir, en el rango entre 1 mW y 5 mW. Como ocurría anteriormente, se aprecia que el requisito de *throughput* objetivo no se cumple. Por ello, se analiza el segundo caso, representado por cajas amarillas, donde se aumenta el rango de interferencia, que se sitúa entre 11 mW y 15 mW. Se comprueba que se sigue sin cumplir el *throughput* objetivo, alejándose aún más que en el caso anterior, pero se aprecia de manera más clara el efecto de los niveles de centralización. En el tercer caso, simbolizado por cajas rosas, se establece una interferencia normal y se reduce la presencia del segundo *slice* a las áreas 8, 9 y 10. Se verifica que ahora sí se alcanzan los 10 Mbps de *throughput* objetivo. En el último caso, representado por cajas verdes, se mantiene el segundo *slice* en las áreas

comentadas, y se incrementa la interferencia, comprobando que el *throughput* aumenta al hacerlo los niveles de centralización y que, dependiendo de estos últimos, se consigue cumplir en ciertos casos el requisito de *throughput* objetivo.



(a) Distribución del throughput medio para los diferentes niveles de centralización y para los distintos escenarios







5.2.2. Normalización de los recursos medios asignados en el tercer escenario

A la vista de los resultados obtenidos anteriormente, se decide estudiar en mayor profundidad el tercer escenario de aplicación, considerando una interferencia comprendida en el rango entre 11 mW y 15 mW, y estableciendo la presencia del segundo *slice* únicamente en las áreas 8, 9 y 10 del escenario. El resto de parámetros de configuración se corresponden con los recogidos en la Tabla 5.1, salvo el nivel de *split*, que cambia su valor, y la complejidad computacional de la CU, que no se tiene en cuenta. Como se muestra en la Figura 5.7, se representan los recursos medios normalizados asignados por cada BS física existente en el escenario a cada uno de los *slices*, en función del nivel de centralización.



Figura 5.7: Recursos medios normalizados asignados a cada *slice* por cada BS física del tercer escenario para los cuatro niveles de centralización

En la Figura 5.7 se observa que, a medida que se aumenta el nivel de centralización, se reduce el porcentaje de los recursos medios asignados por las BS físicas; en otras palabras, se produce una disminución de la cantidad de recursos asignados al incrementar la centralización. Esta gráfica permite visibilizar la ganancia obtenida al incrementar los niveles de centralización, es decir, posibilita la visualización del porcentaje en que se reducen los recursos asignados de un nivel a otro. Además, se aprecia que el segundo *slice* en las tres primeras BSs físicas, y en los niveles de *split 2* y 3, necesita un menor número de recursos que el primer *slice*, debido a que ahora se despliega solo en tres zonas, por lo que no requiere tantos recursos para cumplir ek *throughput* objetivo, al contrario del caso en el que se localizaba en todas las zonas del tercer escenario. Por otro lado, se aprecia que, para el primer *slice* no alcanza el 100 %, y para el caso del segundo nivel de centralización sí. Esto se debe a que, para ambos casos, las BSs físicas saturan y no cuentan con suficientes recursos para asignar, por lo que compensan entre *slices*, es decir, que para el segundo nivel de centralización los recursos que no se asignan al segundo *slice* se dan al primero.

5.3. Estudio de la implementación de niveles de split dinámicos

En esta sección, se evalúa el impacto de realizar una asignación dinámica de los niveles de centralización en las áreas 8, 9 y 10 del tercer escenario, para el segundo *slice*. Para ello, se utiliza el tercer algoritmo implementado, basado en la aproximación *A2*. Además, se mantiene el rango de valores de interferencia más altos (11 mW a 15 mW) y se considera el despliegue del segundo *slice* en las áreas 8, 9 y 10 únicamente. El resto de parámetros de configuración se recogen en la Tabla 5.1. El nivel de *split* del primer *slice* se fija a 1, y el del segundo *slice* cambia en función del valor que se le asigne en cada momento. Adicionalmente, en este caso, sí se utiliza el parámetro de complejidad computacional de la CU, que tomará valores entre 15 y 65.

Esta sección recoge los resultados relacionados con los recursos medios asignados al segundo *slice* en el tercer escenario, en función de la complejidad computacional de la CU considerada, comparándolos con los obtenidos para los niveles de centralización estáticos. Estos resultados se presentan teniendo en cuenta

la eficiencia espectral de 4G y 5G. Por otro lado, se incluyen los resultados relativos a la probabilidad de selección de cada uno de los niveles de centralización en cada una de las tres áreas consideradas, y para todos los valores de complejidad computacional establecidos.

5.3.1. Recursos medios en función de la complejidad computacional

En primer lugar, la Figura 5.8 muestra la distribución de los recursos medios asignados al segundo *slice* entre todas las zonas donde se despliega, en función del valor de la complejidad computacional para dos casos distintos. El primero considera la eficiencia espectral de 4G en el DL, fijada en 5 b/s/Hz, y el segundo la eficiencia espectral de 5G en el DL, que es de 30 b/s/Hz. Además, se representa en ambos casos los recursos medios obtenidos, fijando el nivel de *split* del segundo *slice* a cada uno de los niveles de centralización estáticos.



(a) Recursos medios en función de la complejidad computacional para la eficiencia espectral de 4G

(b) Recursos medios en función de la complejidad computacional para la eficiencia espectral de 5G

Figura 5.8: Distribución de los recursos medios en función de la complejidad computacional para la eficiencia espectral de 4G y de 5G

En la Figura 5.8a se aprecia que, al aumentar el valor de la complejidad computacional de la CU, disminuye el número de recursos medios asignados, debido a que se centraliza un mayor número de funciones y las BSs físicas llevan a cabo menos tareas. También se observa que, si se fija un valor bajo de capacidad computacional, por ejemplo 15, los recursos medios asignados se corresponden con los que se obtendrían si se estableciese un nivel fijo de *split* de 1. En cambio, al incrementar la capacidad computacional, es posible tener combinaciones de *split* más altas en las tres áreas consideradas, lo que permite obtener mejores resultados que con niveles de centralización estáticos. Por otro lado, en el caso de tener una capacidad de cómputo de 65, donde sería posible que las tres áreas presentasen un nivel de centralización máximo, los recursos medios se igualarían a los obtenidos con un nivel estático de *split* de 4. Por lo tanto, en función del valor de la capacidad computacional, se pueden mejorar o igualar los resultados que se obtendrían con niveles de *split* estáticos concretos.

La Figura 5.8b muestra también la distribución de recursos medios en función de la capacidad computacional, pero utilizando una eficiencia espectral de 30 b/s/Hz. Esta eficiencia permite obtener valores mayores de SINR con respecto al caso de la consideración de la eficiencia espectral de 4G, lo que conlleva una reducción notable de los recursos medios asignados, sobre todo para valores altos de

capacidad computacional, donde se pueden asignar niveles de *slit* más altos y, en consecuencia, disminuir más la interferencia y los recursos necesarios. Con respecto al comportamiento observado, se determina que es muy parecido al del caso anterior, con la diferencia de que, a partir de la capacidad de valor 35, y al aplicar una asignación dinámica de niveles de *split*, se mejora en mayor medida los resultados observados, en comparación con los niveles estáticos de centralización. Por otra parte, con la aplicación de la eficiencia espectral de 5G, se consigue reducir notablemente los recursos necesarios con respecto al caso de utilizar la eficiencia espectral de 4G, notando este efecto en mayor medida para valores altos de capacidad computacional.

5.3.2. Probabilidad de selección de cada nivel de split

En la Figura 5.9 se muestra la probabilidad de selección de cada uno de los niveles de *split*, ordenados de menor a mayor centralización, para cada una de las áreas donde se despliega el segundo *slice* y para cada uno de los valores de capacidad computacional de la CU considerados.



Figura 5.9: Probabilidad de selección de cada nivel de split en función del área y de la complejidad computacional de la CU

Como se observa en la Figura 5.9, para valores muy bajos de capacidad de cómputo, en concreto para un valor de 15, se selecciona siempre el primer nivel de *split*, estableciendo la probabilidad de selección del resto de niveles a 0. En el caso contrario, para valores muy altos de capacidad computacional, como por ejemplo el valor 65, se escoge siempre el nivel de centralización 4, debido a que lo mejor es centralizar el mayor número de funciones posibles y para ese valor de complejidad se puede centralizar al máximo en todas las zonas. Para el caso de valores de capacidad computacional intermedios, comprendidos entre 25 y 55, se aprecia que la selección del nivel de *split* ya no es tan estricta, y ahora existe la probabilidad de selección gara más de un nivel. Además, se destaca que, a medida que aumenta el valor de la capacidad computacional de la CU, se incrementa la probabilidad de selección de niveles de centralización mayores, y disminuye la probabilidad de selección de los niveles de *split* más bajos.

Capítulo 6

Conclusiones

En este capítulo se presentan las principales conclusiones obtenidas de la realización de este trabajo. En primer lugar, se comprueba que tanto la implementación del entorno de simulación, como el desarrollo de los algoritmos evaluados, se han finalizado de forma correcta, ya que que todos los resultados obtenidos son coherentes y acordes con lo esperado.

Se ha validado el funcionamiento del primer y segundo algoritmo, verificando que la aplicación de ambos conlleva la estabilidad de las colas virtuales para todas las zonas del primer escenario, y para un *throughput* objetivo del segundo *slice* de 10 Mbps. Por su lado, para valores más altos de este parámetro, su comportamiento es inestable. Después, acotando los resultados para una zona concreta, es decir, para la zona solapada del primer escenario para el segundo *slice*, se ha podido observar que el funcionamiento de ambos algoritmos ha sido adecuado en términos de *throughput* y de estabilidad (colas virtuales). En concreto, se ha determinado que el segundo algoritmo, que aplica la aproximación *A2*, proporciona resultados más estables en términos de ocupación de cola virtual, y cumple con el requisito de *throughput* objetivo en un mayor número de casos que el primer algoritmo.

A continuación, se ha utilizado el segundo algoritmo para obtener el siguiente conjunto de resultados, debido a su mejor funcionamiento. Se han analizado los tres escenarios de aplicación en términos de recursos y *throughput* medios, comprobando que el número de recursos asignados disminuye al aumentar el nivel de centralización, y que el *throughput* cumple su valor objetivo, todo ello para los dos primeros escenarios. Por otro lado, se ha estudiado en mayor profundidad el tercer escenario, variando el nivel de interferencia, constatando el efecto de los niveles de centralización, a la hora de reducir los recursos medios y aumentar el *throughput* medio, cuando se incrementan dichos niveles. Con respecto al *throughput*, se ha verificado el funcionamiento adecuado del segundo algoritmo en el tercer escenario, variando el rango de interferencia y la localización del segundo *slice*.

Por otro lado, se ha comprobado que los resultados más interesantes se obtienen estableciendo un mayor rango de interferencia, para notar el efecto de los niveles de *split*, y limitando la presencia del segundo *slice* a las áreas 8, 9 y 10, de forma que se cumpla el requisito de *throughput* objetivo. Para este caso, se ha verificado que el porcentaje de recursos medios asignados a cada *slice* por cada BS física (tercer escenario) presenta un comportamiento coherente, siendo mayor la reducción de recursos asignados, al incrementar los niveles de centralización.

Finalmente, se ha estudiado la implementación de la asignación de los niveles de centralización de

forma dinámica, mediante el tercer algoritmo, confirmando que se consigue obtener mejores resultados, es decir, que reducen significativamente los recursos medios a asignar, con respecto al caso en el que se establecen los niveles de *split* de forma estática. Al comparar los recursos medios asignados para el caso de las eficiencias espectrales de 4G y 5G, se constata que la de 5G posibilita una mayor reducción de los mismos debido a que, en ese caso, se puede obtener una mayor SINR. Además, se ha analizado la probabilidad de selección de cada nivel de *split* de forma dinámica, concluyendo que, cuanto más altos son los valores que toma la complejidad computacional de la CU, mayor es la probabilidad de selección de niveles más altos.

6.1. Líneas futuras

El desarrollo de este trabajo, centrado en la minimización de los recursos asignados y en la selección de los niveles de *split* adecuados de forma dinámica, tiene continuidad alrededor de varias líneas futuras de investigación, comentadas a continuación:

- La primera se basa en analizar el impacto de aplicar todos los niveles de centralización definidos
 por el 3GPP. Este trabajo se ha centrado en los cuatro niveles más importantes, caracterizando su
 comportamiento en términos de recursos y de *throughput* medios. Se propone ampliar el mismo
 análisis a los ocho niveles existentes.
- La segunda línea de investigación se centra en la ampliación del análisis de la selección de niveles de centralización dinámicos a un mayor número de zonas. Este trabajo ha comprobado el impacto de establecer diferentes niveles de *split* dinámicos en las áreas 8, 9 y 10 del tercer escenario para el segundo *slice*. Se propone ampliar el número de áreas en las que estudiar el impacto de dicha selección, y aplicar la asignación de niveles de centralización dinámicos al primer *slice*.
- La tercera línea consiste en la utilización de escenarios más específicos para la evaluación de los algoritmos desarrollados. Este trabajo ha validado el funcionamiento de los algoritmos para tres escenarios de aplicación distintos. Se propone la modificación de dichos escenarios, añadiendo más BSs físicas y variando el número de recursos que presentan, así como la creación de nuevos escenarios para evaluar situaciones específicas.

Bibliografía

- [1] P. Kulshreshtha y A. K. Garg. "Managing 5G Networks A Review of FSO Challenges and Solutions". En: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 2020, págs. 1-4. DOI: 10.1109/ICCCNT49239.2020.9225591.
- [2] A. Gupta y R. K. Jha. "A Survey of 5G Network: Architecture and Emerging Technologies". En: *IEEE Access* 3 (2015), págs. 1206-1232. DOI: 10.1109/ACCESS.2015.2461602.
- [3] L. M. P. Larsen, A. Checko y H. L. Christiansen. "A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks". En: *IEEE Communications Surveys & Tutorials* 21.1 (2019), págs. 146-172. DOI: 10.1109/COMST.2018.2868805.
- [4] O. Arouk, T. Turletti, N. Nikaein y K. Obraczka. "Cost Optimization of Cloud-RAN Planning and Provisioning for 5G Networks". En: 2018, págs. 1-6. DOI: 10.1109/ICC.2018.8422744.
- [5] Q. K. Ud Din Arshad, A. U. Kashif e I. M. Quershi. "A Review on the Evolution of Cellular Technologies". En: 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST). 2019, págs. 989-993. DOI: 10.1109/IBCAST.2019.8667173.
- [6] A. F. M. Shahen Shah. "A Survey From 1G to 5G Including the Advent of 6G: Architectures, Multiple Access Techniques, and Emerging Technologies". En: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). 2022, págs. 1117-1123. DOI: 10.1109/ CCWC54503.2022.9720781.
- [7] R. Gavrić, D. Ilišević, N. B. Ćurguz y đ. Budimir. "Comparison of basic characteristics of 4G/LTE and 5G NR technology". En: 2019 27th Telecommunications Forum (TELFOR). 2019, págs. 1-4. DOI: 10.1109/TELFOR48224.2019.8971038.
- [8] ITU-R. IMT Vision Framework and overall objectives of the future development of IMT for 2020 and beyond. Inf. téc. M.2083-0. 2015. URL: https://www.itu.int/dms_pubrec/itur/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf.
- [9] M. Tahir, M. H. Habaebi, M. Dabbagh, A. Mughees, A. Ahad y K. I. Ahmed. "A Review on Application of Blockchain in 5G and Beyond Networks: Taxonomy, Field-Trials, Challenges and Opportunities". En: *IEEE Access* 8 (2020), págs. 115876-115904. DOI: 10.1109/ACCESS. 2020.3003020.
- [10] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz y J. M. Lopez-Soler. "A Survey on 5G Usage Scenarios and Traffic Models". En: *IEEE Communications Surveys* & *Tutorials* 22.2 (2020), págs. 905-929. DOI: 10.1109/COMST.2020.2971781.
- [11] ITU-R. Minimum requirements related to technical performance for IMT-2020 radio interface(s). Inf. téc. M.2410-0. 2017. URL: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf.
- R. Gai, X. Du, S. Ma, N. Chen y S. Gao. "A review of 5G applications and key technoligies". En: 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). 2021, págs. 570-574. DOI: 10.1109/ICBAIE52039.2021. 9389982.
- [13] V. Chauhan y Srinivasans. "A Review on 5G Network System with its limitation and different Approaches to build strong 5G Network System". En: 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM). 2022, págs. 403-410. DOI: 10.1109/ICIEM54221. 2022.9853134.
- [14] R. Dilli. "Analysis of 5G Wireless Systems in FR1 and FR2 Frequency Bands". En: 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). 2020, págs. 767-772. DOI: 10.1109/ICIMIA48430.2020.9074973.
- [15] 5G System Overview. https://www.3gpp.org/technologies/5g-system-overview.
- [16] M. A. Habibi, M. Nasimi, B. Han y H. D. Schotten. "A Comprehensive Survey of RAN Architectures Toward 5G Mobile Communication System". En: *IEEE Access* 7 (2019), págs. 70371-70421. DOI: 10.1109/ACCESS.2019.2919657.
- [17] H. Mrabet. "Performance Investigation of New Waveforms in CRAN Architecture for 5G Communication Systems". En: 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS). 2020, págs. 1-5. DOI: 10.1109/ICCAIS48893.2020.9096676.
- [18] Z. Guizani y N. Hamdi. "CRAN, H-CRAN, and F-RAN for 5G systems: Key capabilities and recent advances". En: International Journal of Network Management 27.5 (2017). e1973 nem.1973, e1973. DOI: https://doi.org/10.1002/nem.1973. URL: https://onlinelibrary. wiley.com/doi/abs/10.1002/nem.1973.
- [19] M. Peng, C. Wang, V. Lau y H. V. Poor. "Fronthaul-constrained cloud radio access networks: insights and challenges". En: *IEEE Wireless Communications* 22.2 (2015), págs. 152-160. DOI: 10.1109/MWC.2015.7096298.
- [20] M. Labana y W. Hamouda. "Advances in CRAN Performance Optimization". En: *IEEE Network* 35.3 (2021), págs. 140-146. DOI: 10.1109/MNET.011.2000502.
- [21] A. M. Alba, J. H. G. Velásquez y W. Kellerer. "An adaptive functional split in 5G networks". En: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, págs. 410-416. DOI: 10.1109/INFCOMW.2019.8845147.
- [22] A. M. Alba, S. Janardhanan y W. Kellerer. "Enabling Dynamically Centralized RAN Architectures in 5G and Beyond". En: *IEEE Transactions on Network and Service Management* 18.3 (2021), págs. 3509-3526. DOI: 10.1109/TNSM.2021.3071975.

- [23] A. Martinez Alba y W. Kellerer. "A Dynamic Functional Split in 5G Radio Access Networks". En: 2019 IEEE Global Communications Conference (GLOBECOM). 2019, págs. 1-6. DOI: 10.1109/ GLOBECOM38437.2019.9013336.
- [24] U. Pawar, A. K. Singh, K. Malde, B. R. Tamma y A. Antony Franklin. "Understanding Energy Consumption of Cloud Radio Access Networks: an Experimental Study". En: 2020 IEEE 3rd 5G World Forum (5GWF). 2020, págs. 407-412. DOI: 10.1109/5GWF49715.2020.9221114.
- [25] J. Perälä, P. Jurmu y J. Pinola. "Hybrid Approach for Protocol Testing of LTE System: A Practical Case Study". En: 2010 Second International Conference on Advances in System Testing and Validation Lifecycle. 2010, págs. 32-36. DOI: 10.1109/VALID.2010.18.
- [26] B. Dhindsa, A. Kaur y S. Ahuja. "LTE interfaces and protocols". En: 2015 International Conference on Advances in Computer Engineering and Applications. 2015, págs. 870-874. DOI: 10.1109/ ICACEA.2015.7164827.
- [27] 3GPP. Study on new radio access technology: Radio access architecture and interfaces. Technical Report (TR) 38.801. Version 14.0.0. 2017.
- [28] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo y R. Jain. "Network Slicing for 5G: Challenges and Opportunities". En: *IEEE Internet Computing* 21.5 (2017), págs. 20-27. DOI: 10.1109/MIC.2017.3481355.
- [29] Z. S. Attarbashi, A. E. Altaleb, A. H. Mohd Aman, A. Hassan Abdalla Hashim y S. Eker. "A Review of 5G Technology: Architecture and Challenges". En: 2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED). 2022, págs. 1-4. DOI: 10.1109/ICCED56140. 2022.10010607.
- [30] G. Rahmanian, H. S. Shahhoseini y A. H. J. Pozveh. "A Review of Network Slicing in 5G and Beyond: Intelligent Approaches and Challenges". En: 2021 ITU Kaleidoscope: Connecting Physical and Virtual Worlds (ITU K). 2021, págs. 1-8. DOI: 10.23919/ITUK53220.2021.9662097.
- [31] F. Debbabi, R. Jmal, L. C. Fourati y A. Ksentini. "Algorithmics and Modeling Aspects of Network Slicing in 5G and Beyonds Network: Survey". En: *IEEE Access* 8 (2020), págs. 162748-162762. DOI: 10.1109/ACCESS.2020.3022162.
- [32] *Small cell virtualization functional splits and use cases*. Inf. téc. Document 159.07.02. Small Cell Forum. 2016.
- [33] B. Debaillie, C. Desset y F. Louagie. "A Flexible and Future-Proof Power Model for Cellular Base Stations". En: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring). 2015, págs. 1-7. DOI: 10.1109/VTCSpring.2015.7145603.
- [34] Y. Xiao, J. Zhang e Y. Ji. "Can Fine-Grained Functional Split Benefit to the Converged Optical-Wireless Access Networks in 5G and Beyond?" En: *IEEE Transactions on Network and Service Management* 17.3 (2020), págs. 1774-1787. DOI: 10.1109/TNSM.2020.2995844.
- [35] J. K. Chaudhary, A. Kumar, J. Bartelt y G. Fettweis. "C-RAN Employing xRAN Functional Split: Complexity Analysis for 5G NR Remote Radio Unit". En: 2019 European Conference on Networks and Communications (EuCNC). 2019, págs. 580-585. DOI: 10.1109/EuCNC.2019.8801953.

[36] C. Desset, B. Debaillie, V. Giannini, A. Fehske, G. Auer, H. Holtkamp, W. Wajda, D. Sabella, F. Richter, M. J. Gonzalez, H. Klessig, I. Gódor, M. Olsson, M. A. Imran, A. Ambrosy y O. Blume. "Flexible power modeling of LTE base stations". En: 2012 IEEE Wireless Communications and Networking Conference (WCNC). 2012, págs. 2858-2862. DOI: 10.1109/WCNC.2012. 6214289.