



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# SimuSplit: Simulación para la Gestión Dinámica de Splits y Optimización de Recursos en Redes 5G/6G

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

Valentín Brienza, Marcos Rapetti

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

## TUTOR

Alberto Castro..... Universidad de la República  
Claudina Rattaro..... Universidad de la República  
Lucas Inglés ..... Universidad de la República

## TRIBUNAL

Pablo Bertrand..... Universidad de la República  
Federico La Rocca..... Universidad de la República  
Matías Richart..... Universidad de la República  
Carlos Astudillo..... Universidade Estadual de Campinas

Montevideo  
viernes 17 enero, 2025

*SimuSplit: Simulación para la Gestión Dinámica de Splits y Optimización de Recursos en Redes 5G/6G*, Valentín Brienza, Marcos Rapetti.

Esta tesis fue preparada en L<sup>A</sup>T<sub>E</sub>X usando la clase iietesis (v1.1).  
Contiene un total de 106 páginas.  
Compilada el viernes 17 enero, 2025.

# Resumen

Las redes móviles son fundamentales para las comunicaciones humanas, facilitando la interacción social, el acceso a la educación, la economía global y los servicios críticos. Estas redes están en constante evolución: mientras las redes 5G son una realidad en muchos países, ya se están sentando las bases para la transición hacia las redes 6G. En este contexto, surge OpenRAN, una arquitectura que promueve la interoperabilidad entre diferentes fabricantes mediante componentes desagregados y virtualizados, conectados a través de interfaces abiertas y estándares definidos. Tanto la 3GPP (*3rd Generation Partnership Project*) como OpenRAN prevén configuraciones variadas que permitan elegir distintos puntos de *split* funcionales en las redes de acceso.

Los puntos de *split* funcionales dividen las tareas de procesamiento en diferentes nodos de la red, como la unidad de radio (RU) y las unidades centralizadas o distribuidas (CU/DU). La 3GPP define 8 puntos de *splits* diferentes, pero cada uno implica compromisos: un *split* alto reduce la carga de datos en los enlaces, pero requiere mayor capacidad de procesamiento local; mientras que un *split* bajo centraliza el procesamiento, aumentando la latencia y la demanda de ancho de banda en la red de interconexión; esta en general es una red óptica. La selección óptima depende del estado de la red y de las necesidades específicas del servicio, como fiabilidad, latencia y capacidad. Esta flexibilidad es clave para satisfacer la creciente demanda de aplicaciones avanzadas como *cloud computing*, *streaming* 8K e Internet de las cosas (IoT).

Este proyecto se enmarca en la línea de investigación del proyecto CSIC I+D “Convergencia Óptico-Móvil para Redes Avanzadas”, que aborda múltiples ejes relacionados con el diseño y la optimización de estas redes. Entre ellos se incluyen el desarrollo de nuevas arquitecturas de monitoreo, algoritmos de aprendizaje automático para la asignación de recursos dinámicos y la experimentación con plataformas de simulación. Estas actividades combinan teoría y práctica, utilizando tanto simuladores como testbeds para validar los enfoques propuestos. En este contexto, nuestro proyecto de fin de carrera busca contribuir al diseño y evaluación de arquitecturas óptico-móviles avanzadas.

En este proyecto de grado, se desarrolló una herramienta de simulación basada en bibliotecas de código abierto para OMNeT++, enfocada en la implementación de puntos de *split* funcionales definidos por la 3GPP. Se decidió implementar 3 de ellos: un *split* bajo (*split* 2: entre capas PDCP y RLC), uno medio (*split* 4: entre capas RLC y MAC) y uno alto (*split* 6: entre capas MAC y PHY). Al mismo

tiempo se diseñó una nueva entidad que implementa estos *splits* y que permite gestionar dinámicamente la configuración del punto de *split* durante la simulación. Para evaluar la herramienta, se consideraron distintos escenarios de interés para las redes 5G, incluyendo topologías con multidominio tecnológico (óptico-móvil), y se analizó el efecto que tiene el punto de *split* en métricas como el retardo.

# Acrónimos

- **3GPP:** 3rd Generation Partnership Project
- **3G:** Tercera generación de redes móviles
- **4G:** Cuarta generación de redes móviles
- **5G:** Quinta generación de redes móviles
- **5G-CN:** 5G Core Network
- **5G-NR:** 5G New Radio
- **5G-NSA:** 5G Non Standalone
- **5G-SA:** 5G Standalone
- **ACK:** Acknowledge
- **AF:** Application Function
- **AM:** Acknowledged Mode
- **AMC:** Adaptive Modulation and Coding
- **AMF:** Access and Mobility Management Function
- **ARQ:** Automatic Repeat reQuest
- **AUSF:** Authentication Server Function
- **BBU:** Baseband Unit
- **BGP:** Border Gateway Protocol
- **CBR:** Constant Bit Rate
- **CN:** Core Network
- **CPRI:** Common Public Radio Interface
- **CPU:** Central Processing Unit
- **CQI:** Channel Quality Indicator

- **C-RAN:** Cloud-RAN o Centralized-RAN
- **CSI:** Channel State Information
- **CSIC:** Consejo Superior de Investigaciones Científicas
- **CU:** Central Unit
- **D2D:** Device-to-device
- **DL:** Downlink
- **DT:** Digital Twin
- **DU:** Distributed Unit
- **DWDM:** Dense WDM
- **eCPRI:** Enhanced CPRI
- **EON:** Elastic Optical Network
- **eMBB:** Enhanced Mobile Broadband
- **eNB:** Evolved Node B
- **EPC:** Evolved Packet Core
- **E-ULTRA:** Evolved UMTS Terrestrial Radio Access
- **FDD:** Frequency Division Duplex
- **gNB:** Next Generation Node B
- **GPRS:** General Packet Radio Service
- **GTP:** GPRS Tunneling Protocol
- **HARQ:** Hybrid-ARQ
- **IDE:** Integrated Development Environment
- **IoT:** Internet of Things
- **IP:** Internet Protocol
- **IP2NIC:** IP-to-NIC
- **IPv4:** Internet Protocol version 4
- **IPv6:** Internet Protocol version 6
- **LTE:** Long Term Evolution
- **MAC:** Medium Access Control

- **MCS:** Modulation Coding Scheme
- **MEC:** Multi-access Edge Computing
- **ML:** Machine Learning
- **mMTC:** Massive Machine-Type Communication
- **MUX:** Multiplexer
- **NACK:** Negative Acknowledge
- **NAS:** Non-Access Stratum
- **NED:** Network Description
- **NEF:** Network Exposure Function
- **nFAPI:** Network Functional Application Platform Interface
- **NIC:** Network Interface Card
- **NR:** New Radio
- **NRF:** Network Repository Function
- **NS-3:** Network Simulator 3
- **NSA:** Non Standalone
- **OAI:** Open Air Interface
- **OFDM:** Orthogonal Frequency Division Multiplexing
- **OMNeT++** Objective Modular Network Testbed in C++
- **OSI:** Open Systems Interconnection
- **OSPF:** Open Shortest Path First
- **P2P:** Point-to-point
- **P2MP:** Point-to-multipoint
- **PCF:** Policy Control Function
- **PDCP:** Packet Data Convergence Protocol
- **PHY:** Physical Layer
- **PPP:** Point-to-point Protocol
- **RAM:** Random Access Memory
- **RAN:** Radio Access Network

- **RB:** Resource Block
- **RF:** Radio Frequency
- **RLC:** Radio Link Control
- **ROHC:** Robust Header Compression
- **RRC:** Radio Resource Control
- **RRH:** Remote Radio Head
- **RRU:** Remote Radio Unit
- **RTP:** Real-time Transport Protocol
- **RU:** Radio Unit
- **RX:** Recepción
- **SCTP:** Stream Control Transmission Protocol
- **SINR:** Signal to Interference Plus Noise Ratio
- **SM:** Split Manager
- **SMF:** Session Management Function
- **TB:** Transport Block
- **TBS:** Transport Block Size
- **TCP:** Transmission Control Protocol
- **TDD:** Time Division Duplex
- **TDM:** Time Division Multiplexing
- **TEID:** Tunneling Endpoint Identifier
- **TM:** Transparent Mode
- **TTI:** Transmission Time Interval
- **TWDM:** Time and Wavelength Division Multiplexing
- **TX:** Transmisión
- **UDM:** Unified Data Management
- **UDP:** User Datagram Protocol
- **UE:** User equipment
- **UL:** Uplink

- **UM:** Unacknowledged Mode
- **UMTS:** Universal Mobile Telecommunications System
- **UPF:** User Plane Function
- **URLLC:** Ultra-Reliable and Low Latency Communication
- **VoIP:** Voice over Internet Protocol
- **WDM:** Wavelength Division Multiplexing
- **WSL:** Windows Subsystem for Linux

Esta página ha sido intencionalmente dejada en blanco.

# Tabla de contenidos

<b>Resumen</b>	<b>I</b>
<b>Acrónimos</b>	<b>III</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Antecedentes . . . . .	4
1.3. Objetivos . . . . .	6
1.4. Estructura del documento . . . . .	6
<b>2. Marco Teórico</b>	<b>9</b>
2.1. Redes 5G . . . . .	9
2.1.1. Núcleo de la red . . . . .	11
2.1.2. Red de acceso . . . . .	13
2.1.3. Comunicación entre el usuario y el núcleo . . . . .	17
2.1.4. Splits funcionales . . . . .	18
2.2. Redes ópticas . . . . .	25
2.2.1. Integración óptico-eléctrica . . . . .	26
2.2.2. Grilla flexible . . . . .	26
<b>3. Simuladores</b>	<b>29</b>
3.1. Elección del simulador . . . . .	29
3.2. OMNeT++ . . . . .	30
3.3. INET . . . . .	32
3.4. Simu5G . . . . .	33
3.5. Optical Network Simulator . . . . .	36
<b>4. Implementación</b>	<b>39</b>
4.1. Biblioteca Simusplit . . . . .	39
4.2. Split Manager . . . . .	39
4.2.1. Conexión de múltiples DU a una CU . . . . .	44
4.2.2. Capacidad de procesamiento . . . . .	44
4.2.3. Opciones de <i>split</i> implementadas . . . . .	45
4.3. Módulo DU/CU . . . . .	46
4.3.1. Mensajes SMControl . . . . .	47
4.3.2. Parámetros . . . . .	47

## Tabla de contenidos

<b>5. Experimentación</b>	<b>49</b>
5.1. Topologías . . . . .	49
5.2. Tipos de Tráfico . . . . .	50
5.3. Simulaciones . . . . .	53
5.3.1. Parámetros Iniciales . . . . .	53
5.3.2. Validación con Simu5G . . . . .	54
5.3.3. Variación de cantidad de usuarios . . . . .	54
5.3.4. Tiempo de procesamiento de los equipos . . . . .	54
5.3.5. Interferencia de otras radiobases . . . . .	54
5.3.6. Variación del tamaño de paquete . . . . .	55
5.3.7. Comparación de la distribución de usuarios en cada DU . . . . .	55
5.3.8. Comparación de cantidad de usuarios en distintas celdas conectadas mediante red óptica . . . . .	55
5.4. Estadísticas . . . . .	56
<b>6. Análisis de resultados</b>	<b>59</b>
6.1. Topología 0 . . . . .	59
6.2. Topología 1 . . . . .	59
6.2.1. Comparación de cantidad de usuarios . . . . .	61
6.2.2. Interferencia de otras radiobases . . . . .	63
6.2.3. Variación del tamaño del paquete . . . . .	66
6.3. Topología 2 . . . . .	67
6.3.1. Comparación de distribución de usuarios por celda . . . . .	67
6.4. Topología 3 . . . . .	69
6.4.1. Comparación de la cantidad de usuarios . . . . .	69
<b>7. Conclusiones y trabajo futuro</b>	<b>71</b>
7.1. Herramienta obtenida . . . . .	71
7.2. Trabajo a futuro . . . . .	72
7.2.1. Extender las opciones de <i>split</i> soportadas . . . . .	72
7.2.2. Asignación dinámica del punto del <i>split</i> . . . . .	72
7.2.3. Protocolos en el enlace DU-CU . . . . .	73
7.2.4. Ajuste de parámetros . . . . .	73
7.2.5. Separación de punto de <i>split</i> según cliente . . . . .	73
7.2.6. Impacto de red óptica en puntos de <i>split</i> . . . . .	73
7.2.7. Simulación de canal satelital . . . . .	74
<b>Apéndices</b>	<b>75</b>
<b>A. Cambios en Simu5G</b>	<b>75</b>
A.1. Nuevas puertas . . . . .	75
A.1.1. Comportamiento RLC . . . . .	75
A.2. Dirección de los Mensajes . . . . .	76
A.3. GtpUser / TxPDCP (UE DestId) . . . . .	77
A.4. Comportamiento IP2NIC para los UE no conectados al CU . . . . .	77
A.5. MAC . . . . .	77

<b>B. Scripts</b>	<b>79</b>
B.1. Recolección de estadísticas . . . . .	79
B.2. <code>get_results.py</code> . . . . .	79
B.3. <code>run_mixed.py</code> . . . . .	80
B.4. Analisis de resultados . . . . .	80
<b>C. Medidas de <i>performance</i></b>	<b>81</b>
C.1. Ambiente de Simulación . . . . .	81
C.2. Tiempo de ejecución de las simulaciones . . . . .	81
<b>D. Gráficas</b>	<b>83</b>
<b>Referencias</b>	<b>87</b>

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 1

## Introducción

En esta sección se presentará la importancia de estudiar y desarrollar herramientas que implementen los *splits* funcionales. Así mismo, se muestra que este tipo de tecnologías están siendo estudiadas actualmente por la comunidad científica. Por último, se esboza cómo será organizado el documento.

### 1.1. Motivación

La evolución de las redes móviles históricamente se ha caracterizado por hardware de propósito específico, con implementaciones de las funciones de red atadas fuertemente al fabricante. Esto implica grandes costos a la hora desplegar nuevas tecnologías ya que se deben adquirir equipos distintos de los ya existentes, lo que a su vez introduce demoras en la adopción de nuevos estándares. Además dificulta la escalabilidad de la red, ya que típicamente no existe interoperatividad entre equipos de distinto fabricante, limitando la capacidad de los operadores de redes móviles de elegir aquel que mejor se ajuste a sus necesidades.

Por otra parte, el uso que los clientes le dan a la red móvil ha ido evolucionado y cambiando a través de las generaciones. En sus inicios estaban enfocadas exclusivamente a la comunicación de voz. Posteriormente, con el auge de Internet y los avances tecnológicos asociados a los dispositivos móviles, el uso de la red se diversificó a aplicaciones con una mayor demanda de ancho de banda como videos y juegos en línea. En los últimos años, la novedad ha venido por nuevos tipos de servicios, donde el usuario ya no representa un ser humano, sino un dispositivo como en el caso de Internet de las Cosas (IoT). Estos dispositivos demandan otras características de la red, como una mayor confiabilidad para el monitoreo de procesos críticos o baja latencia para tomar acciones en tiempo real.

Para adaptarse a estas tendencias, los operadores de redes móviles han puesto el interés en arquitecturas de red más flexibles y desagregadas con el objetivo de reducir costos y mejorar la eficiencia de la red [29]. Una de las novedades introducidas en 4G, y potenciada en 5G que posibilita esta flexibilidad, es la capacidad de dividir las funcionalidades que proveen el acceso a los usuarios en nodos de la red distintos. Tradicionalmente, se encontraban junto a la antena los equipos

## Capítulo 1. Introducción

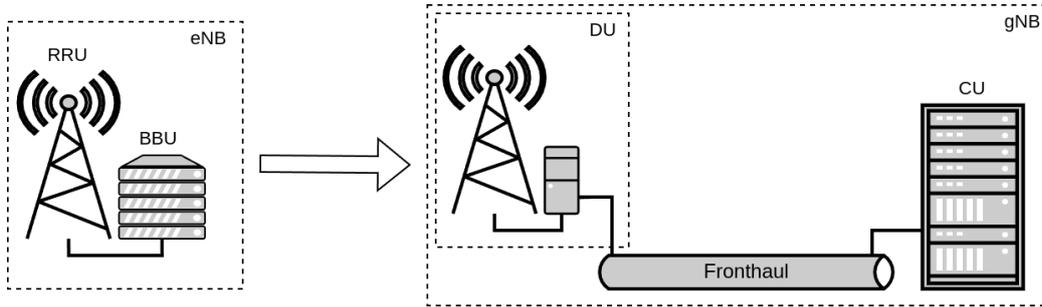


Figura 1.1: A la izquierda se ilustra la arquitectura tradicionalmente utilizada donde todas las funciones para proveer acceso a los usuarios se encuentran localizados en un mismo sitio. A la derecha se ilustra la arquitectura utilizando *splits* funcionales en la que se encuentra la antena y ciertas funcionalidades en un sitio y el resto del procesamiento centralizado en otro.

encargados de las tareas de recepción/transmisión (RX/TX) y procesamiento de las señales. En la generación previa, 4G, hubo un primer acercamiento a la separación, denominando *Remote Radio Unit* (RRU) al equipo responsable de RX/TX, *Baseband Unit* (BBU) al del procesamiento y *Evolved NodeB* (eNB) al conjunto de estos dos equipos. Sin embargo, como el BBU realiza la totalidad del procesamiento, ubicarlo en un sitio alejado del RRU presenta desafíos en términos de la latencia introducida por el enlace entre ambos. Al mismo tiempo esto requiere un alto ancho de banda en la conexión entre ambos equipos.

En 5G se flexibiliza esta separación de tareas introduciendo el concepto de *splits* funcionales, en donde se define una unidad distribuida (DU) y una centralizada (CU). Esto permite centralizar únicamente algunas funciones según las necesidades, imponiendo menos requisitos sobre el enlace que los comunica. En la Figura 1.1 se observa una comparación de estas dos arquitecturas, la utilizada tradicionalmente en generaciones anteriores concentra la totalidad de las funcionalidades en un mismo elemento denominado *Next Generation NodeB* (gNB). Por otro lado, la arquitectura utilizando *splits* divide este elemento en dos, comunicados a través de un enlace denominado *fronthaul*. El nivel de centralización es establecido en base a las funciones de red ubicadas en cada equipo, en particular se encuentran definidos por la 3GPP (*3rd Generation Partnership Project*) los puntos de *split* ilustrados en la Figura 1.2. Algunos de ellos coinciden con la división de capas ya existentes en generaciones anteriores, mientras que otros dividen funcionalidades específicas de estas. Se resaltan en rojo aquellas opciones que son de interés para este trabajo. Otras estandarizaciones, con ideas similares respecto a dividir la gNB en varios nodos, definen a su vez la unidad de radio (RU) —con similitudes al RRU de la generación previa— donde se encuentran únicamente las funcionalidades asociadas a la capa física. De esta forma, se dividen las funcionalidades de la DU definida por la 3GPP en dos equipos DU y RU.

Una de las ideas principales de este nuevo paradigma implica utilizar hardware genérico para realizar las funciones de red, particularmente en la CU, permitiendo un nivel mayor de flexibilidad y escalabilidad en comparación al hardware propietario utilizado hoy en día. Esta idea se encuentra fuertemente vinculada con

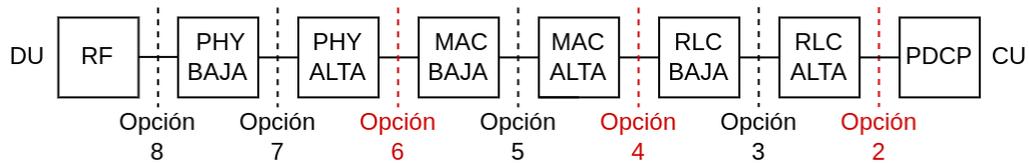


Figura 1.2: Puntos de *split* definidos por 3GPP para la separación de funciones asociadas a la comunicación con los usuarios en 5G. En rojo se encuentran resaltadas las opciones abordadas en este trabajo.

la virtualización de las funciones de red, la cual habilita una gran capacidad de reconfiguración y adaptación de los recursos disponibles en tiempo real [25], así como la distribución de las distintas funciones de red en locaciones geográficamente distintas. La centralización de funciones de red ubicadas tradicionalmente en las radiobases, responsables de dar conexión a los usuarios, permite reducir los costos asociados al despliegue de nuevos puntos de acceso. Además, debido a la menor necesidad de procesamiento en los equipos distribuidos, su tamaño se ve reducido, haciéndolas viables para ciertos escenarios como interiores de edificios o zonas con grandes densidades de usuarios. Como contraparte, la centralización de estas funciones posee ciertos desafíos en términos de latencia y ancho de banda requerido para la conexión entre los distintos elementos de la red. Algunas funciones requieren de un tiempo de respuesta muy bajo con el usuario como puede ser la planificación de los recursos de aire. Del mismo modo, también existen aplicaciones que requieren de esta baja latencia para cumplir su función. Por otro lado, realizar un menor procesamiento en los equipos distribuidos implica que la cantidad de datos que deben ser transmitidos a la unidad central es mayor, lo cual puede saturar la red de transporte utilizada para comunicar los distintos sitios.

Otra tendencia en la evolución de las redes móviles es la incorporación del cómputo en el borde de la red, conocido como *Multi-access Edge Computing* (MEC). La idea principal de MEC consiste en acercar el cómputo y almacenamiento de datos al usuario, de forma de responder a nuevos servicios que requieren baja latencia y alta confiabilidad. Para el soporte de esta nueva arquitectura, se requiere de una red de acceso que permita mantener sus funcionalidades de forma local en el sitio junto a los recursos de cómputo para una rápida respuesta. Para que esto coexista con la centralización de las funciones de red, se espera que los operadores mantengan recursos tanto de forma centralizada y distribuida. De esta manera, se logran satisfacer estos usos específicos pero a su vez se disminuyen los gastos al centralizar el procesamiento para aquellos usuarios que no requieren de esta baja latencia. Por otro lado, también puede ser útil tener los recursos de cómputo centralizados junto al procesamiento de la red en casos donde se deban procesar datos provenientes de múltiples sitios. Esto permite dar un mejor servicio dado que nuevamente se mantiene el procesamiento de la aplicación cerca de la conexión con la red, además de permitir un manejo flexible de los recursos disponibles tanto para dar servicios como para mantener las funciones de red. En la Figura 1.3 se muestra un esquema de la arquitectura de red 5G con MEC, se puede observar la red de acceso dividida en tres elementos RU, DU y CU como

## Capítulo 1. Introducción

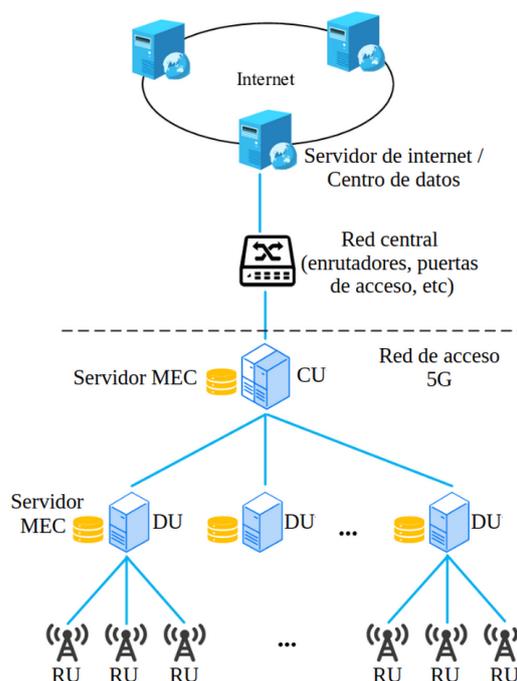


Figura 1.3: Arquitectura de red MEC en la cual junto a los equipos de la red móvil se despliegan servidores con capacidades de computo para ofrecer este servicio a clientes que requieran un tiempo de respuesta reducido. Imagen adaptada de [48].

se mencionó anteriormente, junto a distintos servidores para poder dar servicio de cómputo en el borde de la red a distintos niveles de agregación.

### 1.2. Antecedentes

La simulación de los *splits* funcionales para 5G es un tema que ha cobrado relevancia en los últimos años, impulsado por operadores y vendedores junto con la academia, para contar con herramientas que permitan analizar nuevas arquitecturas o algoritmos. Diversos trabajos se han enfocado en evaluar los beneficios y desafíos en diferentes opciones de *split*.

En [23] se propone un esquema de *split* dinámico, y se plantean algoritmos para la selección del nivel de centralización en base a optimización de la tasa de transmisión sobre un modelo. Los resultados son validados en un simulador modificado basado en MATLAB y C++. Otro trabajo con un enfoque similar es [14], donde se optimiza el ancho de banda necesario para la conexión entre las unidades de procesamiento central y remota. En este caso, la simulación se realiza en un simulador de eventos discretos sobre MATLAB.

En [26] se propone un modelo de optimización para la división de la arquitectura de red de forma de poder atender a distintos tipos de servicios. El modelo utiliza la selección de *split* como uno de los grados de libertad para poder cumplir los recursos de red, enlace y procesamiento disponibles. Para este trabajo se utiliza

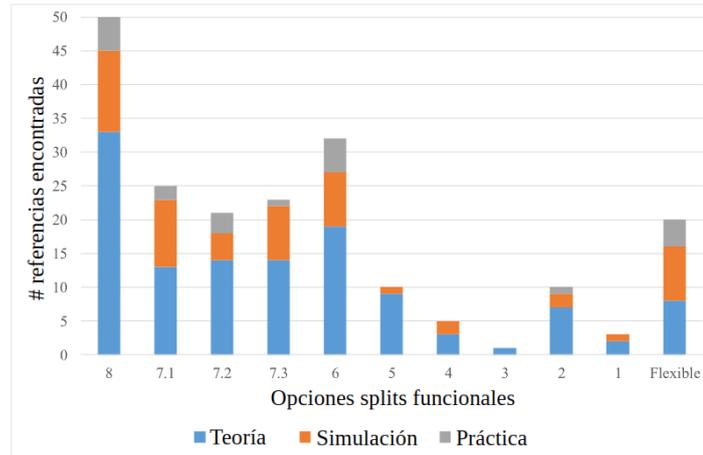


Figura 1.4: Cantidad de referencias encontradas en [22] para las diferentes opciones de *split*, segmentado por investigación teórica, práctica y mediante simulación.

un modelo puramente matemático, en el que se modela el costo de procesamiento asociado a cada función de red, de forma de analizar el impacto de centralizar o no un conjunto de ellas.

Por otra parte, algunos trabajos han utilizado la validación en entornos reales como en [27], donde se utiliza un prototipo de red 5G basada en *Open Air Interface* (OAI). En él se busca optimizar el costo de despliegue de la *Radio Access Network* (RAN), considerando requisitos de cómputo de las funciones y las capacidades de la infraestructura en la nube.

En la Figura 1.4 se muestra la cantidad de referencias encontradas en [22] para las diferentes opciones de *split*. La gran mayoría de ellos se concentran en los *splits* altos (a partir del 6), particularmente en simulación e investigación práctica. Nuestro trabajo busca abarcar opciones de *split* bajos y medios, y de esta forma permitir el estudio de opciones variadas en una misma herramienta.

Los trabajos revisados se encuentran enfocados principalmente en la optimización de la elección del punto de split en base a distintas métricas, y casi la totalidad de ellos validan sus resultados en simulaciones o maquetas. Si bien esta segunda opción es de esperar que provea resultados más realistas, típicamente no es accesible y no permite escalar a escenarios con grandes cantidades de usuarios. En base a estas observaciones, es clara la importancia de contar con una herramienta de simulación que permita evaluar e investigar nuevas implementaciones de funciones en la red. Además, esta necesidad no es específica del ámbito científico sino también para operadores comerciales gracias a que les permite verificar sus cambios previo a implementarlos en su red, disminuyendo su riesgo. Estas herramientas permiten realizar un análisis detallado sobre el impacto de distintas técnicas y tecnologías en una red. A su vez, permiten la creación de *gemelos digitales* (DT) que intentan ser réplicas virtuales de un escenario físico y real. Por un lado estos DT pueden ayudar a operadores en el análisis en tiempo real del estado de la red. Por otro, es posible generar herramientas predictoras de tráfico, que optimicen los recursos y

## Capítulo 1. Introducción

mejorando métricas de interés. En particular, los DT tienen una gran sinergia con técnicas de *Machine Learning* (ML) [11].

Como se puede observar en los trabajos mencionados, una herramienta típicamente utilizada como base para realizar simulaciones es MATLAB, el cual es un software de pago. En su defecto, algunos de estos trabajos utilizan implementaciones propias que de igual manera no se encuentran disponibles de forma abierta. Nuestro trabajo extiende una herramienta de simulación con soporte para 5G de código abierto lo que permite aprovechar la gran cantidad de contribuciones ya realizadas por la comunidad. La solución propuesta permite simular *splits* con diferentes niveles de centralización, en particular las opciones 2, 4 y 6. El diseño elegido para su implementación cuenta con un controlador central en los equipos DU y CU que concentra la lógica asociada al punto de *split*, facilitando la implementación de otras opciones en un futuro sin necesitar grandes modificaciones en la arquitectura de los nodos. Si bien este trabajo no implementó algoritmos de selección dinámica del punto de *split*, este diseño en base a un controlador tuvo en cuenta esta posibilidad para futuros aportes. Finalmente, muchos de los simuladores utilizados en los trabajos mencionados utilizan implementaciones de bajo nivel, que si bien simulan con mayor precisión el comportamiento de las señales transmitidas en capa física, también poseen menor flexibilidad para simular arquitecturas complejas y variadas. El simulador base elegido para este trabajo, OMNeT++, posee implementaciones de capas superiores a las de la red móvil, lo que habilita una representación más precisa de casos de uso específicos a nivel de aplicación.

### 1.3. Objetivos

El objetivo principal de este proyecto es el diseño e implementación de una herramienta de simulación capaz de emular y evaluar distintos puntos de *split*. Se espera que esta herramienta sea versátil, permitiendo ser extendida en un futuro a más puntos de *split*. A su vez, se desea que cuente con la posibilidad de asignar dinámicamente los puntos de *split* durante una simulación y de esta forma permitir evaluar algoritmos para la elección del nivel de centralización.

### 1.4. Estructura del documento

A continuación se verá una breve descripción de la estructura del documento.

En el capítulo 2 se presentan algunos conceptos teóricos del funcionamiento de las redes móviles, las ópticas y los distintos puntos de *splits* definidos por la 3GPP.

El capítulo 3 se centra en introducir las herramientas de simulación que fueron utilizadas como base del desarrollo del proyecto.

Por otra parte, es en el capítulo 4 que se desarrolla cómo fueron implementadas las opciones de los puntos de *split*, y se justifica por qué se eligió implementarlo de esa forma.

El capítulo 5 describe los distintos escenarios de simulación planteados; explicando su topología y parámetros elegidos. También se explican las estadísticas

#### 1.4. Estructura del documento

extraídas y la forma de presentar los resultados. El análisis de estos resultados se encuentra en el capítulo 6.

Por último, en el capítulo 7 se presentan las conclusiones del proyecto y se muestran distintas ramas en las cuales expandir el trabajo realizado.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 2

## Marco Teórico

A continuación se describen los conceptos teóricos básicos necesarios para comprender el funcionamiento y arquitectura de las redes 5G, en particular los aspectos de interés para los *splits* funcionales en 5G. Se describen las tecnologías y protocolos sobre las cuales se realiza la división de las funciones de la gNB. Además se describen brevemente las tecnologías de redes ópticas típicamente utilizadas para la conexión de distintos elementos de la red.

### 2.1. Redes 5G

Desde la aparición de la primera generación de redes móviles en 1980, donde su uso estaba limitado a servicios de voz basados en modulaciones analógicas, estas han crecido continuamente tanto en cantidad de usuarios como en capacidades. Con la segunda generación se incluyó la transmisión digital que, aunque continuaba estando enfocada en llamadas, brindó una mayor versatilidad que hizo posible comunicar otro tipo de datos. La tercera generación o 3G trajo consigo importantes mejoras en esta dirección, permitiendo el acceso a internet con velocidades considerables. Para el desarrollo de esta generación se conformó la 3GPP, quien definió estándares globales para estas tecnologías, ya que aún coexistían distintas estandarizaciones para las redes móviles previas. La próxima generación, 4G, ya ampliamente desplegada en el mundo y en gran actividad, continuó con el avance en las tasas de transferencias alcanzables. 5G, la quinta generación de redes móviles, es la más reciente de estas tecnologías y se encuentra actualmente en despliegue.

Esta nueva tecnología cuenta con mejoras en direcciones similares a las versiones previas, como lo es el incremento en las tasas de transferencia alcanzables, pero además propone otros casos de usos con requerimientos distintos. Estos últimos son en gran parte motivados por las aplicaciones asociadas al despliegue de dispositivos inteligentes como grandes redes de sensores o robots autónomos. Se definen tres grandes clasificaciones para las comunicaciones, ilustradas en la Figura 2.1 donde además se indican algunos requisitos que se esperan para ellas respecto a la tasa de transferencia, latencia, densidad de dispositivos, entre otros. Estas

## Capítulo 2. Marco Teórico

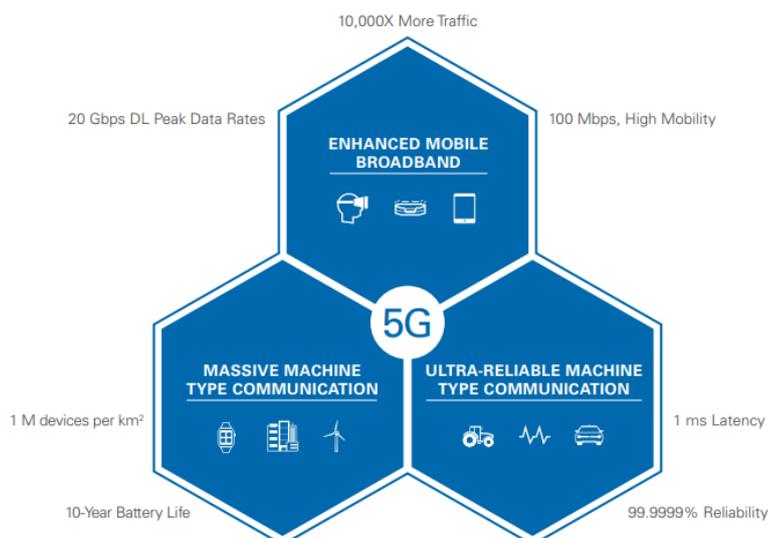


Figura 2.1: Diagrama de casos de uso previstos para 5G, se incluyen cifras ejemplificadoras de los requisitos para cada uno como tasa de transferencia de datos, densidad de dispositivos, latencia, entre otros. Imagen extraída de [8].

clasificaciones son:

- *Enhanced Mobile Broad Band (eMBB)*: Corresponde al uso más tradicional en los últimos años para las redes móviles como lo es la transferencia de datos en grandes volúmenes, y en la que se pretende seguir mejorando la experiencia del usuario.
- *Massive Machine-Type Communication (mMTC)*: Hace referencia a casos en los que se tiene una gran cantidad de dispositivos transmitiendo cada uno, en su gran mayoría, una pequeña cantidad de datos. La importancia de este caso de uso viene dada por el auge de lo que se conoce como IoT, en la que sensores, controles, medidores, y otros dispositivos se encuentran conectados a internet permitiéndoles transferir datos y trabajar en conjunto. Luego una de las características que se busca lograr desde el lado de la red móvil es permitir que los transmisores/receptores de estos dispositivos sean de bajo costo y con un consumo de energía bajo dado que muchos de ellos podrían funcionar con baterías.
- *Ultra-Reliable and Low Latency Communication (URLLC)*: Como su nombre indica, esta contempla dispositivos que necesiten baja latencia y alta confiabilidad. Los autos autónomos y automatizaciones en fabricas son las aplicaciones más frecuentemente utilizadas para ejemplificar esta clase.

Una red móvil 5G es formada por dos secciones principales, denominadas la red de acceso (RAN) y el núcleo de la red (CN) conectadas a través de una red de transporte. La RAN es responsable de la comunicación de forma inalámbrica entre el usuario y la red, mientras que el núcleo de la red es responsable de las

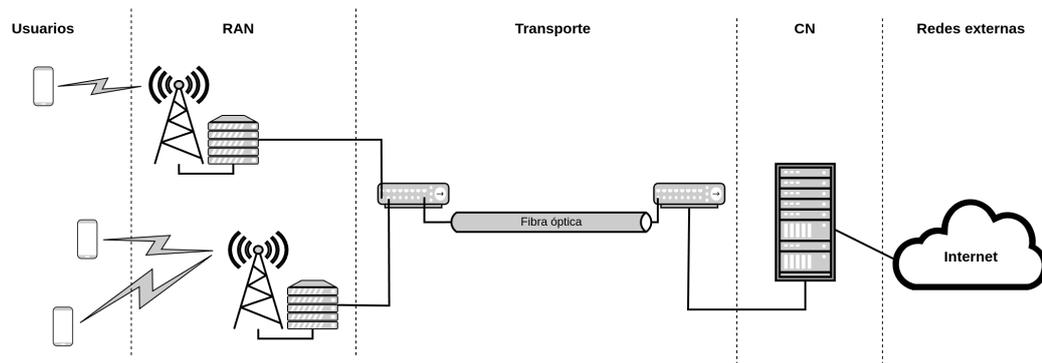


Figura 2.2: Diagrama de los distintos segmentos en una red móvil 5G y sus elementos como la red de acceso que da conexión a los usuarios, el núcleo de la red que provee comunicación con redes externas y la red de transporte que conecta estos elementos. Se ejemplifican algunos de ellos con usos comunes como la conexión con internet o la red de transporte óptica.

funcionalidades asociadas a proveer una red completa manteniendo bases de datos de los usuarios y realizando funciones de enrutamiento dentro de la red así como con redes externas. En la Figura 2.2 se ilustran las distintas secciones de la red, utilizando casos usuales para algunos de ellos como es una red de transporte óptica o conexión con Internet como red externa. En los siguientes apartados se describen las características y componentes de ambas partes de la red.

### 2.1.1. Núcleo de la red

*5G Core Network* (5G-CN) fue diseñado en base a tres ideas fundamentales respecto al *Evolved Packet Core* (EPC) de la anterior generación: arquitectura basada en servicios, soporte para *network slicing* y mayor énfasis en la separación del plano de control y plano de usuario.

La arquitectura basada en servicios implica que el núcleo de la red es definido por las diferentes funcionalidades que provee en lugar de nodos físicos. Este diseño sigue las tendencias de los operadores de virtualizar la red, donde el núcleo de la red es desplegado en servidores de propósito general en lugar de equipos dedicados. Esta iniciativa es conocida como *Telco Cloud* y permite a las empresas de telecomunicaciones desplegar o escalar funciones de forma más sencilla y dinámica, lo que facilita mantener una red actualizada [44].

Otra innovación para brindar servicios a terceros es la incorporación del concepto de *Network slicing*. Esta es una tecnología que permite dividir la red en múltiples redes virtuales, cada una con sus propias características y funcionalidades pero utilizando el mismo despliegue de infraestructura por parte del operador. Esto permite garantizar los requerimientos de los distintos casos de uso, como por ejemplo la baja latencia para los autos autónomos, o la alta tasa de transferencia para el tráfico de internet. La división no abarca únicamente los recursos de la red móvil sino a la arquitectura por completo, incluyendo la red de transporte, lo que permite asegurar los requisitos ofrecidos a los clientes de extremo a extremo de la red.

## Capítulo 2. Marco Teórico

Finalmente, el núcleo de 5G busca separar las funciones del plano de usuario del plano de control. En primera instancia esto permite escalar ambas funcionalidades de forma aislada para adaptarse a las demandas actuales. Por otro lado, la separación de estas funciones también permite una localización de estas de forma conveniente para los distintos casos de uso. Por ejemplo, ubicar el plano de usuario cerca del usuario permite reducir la latencia en la conexión con otras redes. Mientras que el plano de control puede ser centralizado para disminuir costos, ya que estas funciones usualmente permiten mayor flexibilidad de latencias por parte de los usuarios.

En la Figura 2.3 se muestra un diagrama de la arquitectura de red de 5G, donde se pueden observar las distintas funciones y su interacción. A continuación se menciona en forma resumida las funcionalidades de cada uno:

- *User Plane Function* (UPF): Es la interfaz entre la red móvil y las redes externas, como internet. Aquí se realizan las funciones de enrutamiento, inspección y filtrado de paquetes, así como la gestión de la calidad de servicio asociada a cada usuario.
- *Session Management Function* (SMF): Esta función se encarga de mantener las sesiones asociadas a cada usuario, como por ejemplo la asignación de direcciones IP.
- *Access and Mobility Management Function* (AMF): Es la encargada de la de la señalización entre el núcleo y el usuario, así como el cambio de celdas cuando el usuario se encuentra inactivo.
- *Authentication Server Function* (AUSF): Provee las funcionalidades asociadas a la autenticación y autorización de los usuarios.
- *Policy Control Function* (PCF): Realiza el control de las políticas de la red definidas.
- *Unified Data Management* (UDM): Se encarga de la gestión de la información de los usuarios como sus credenciales.
- *Network Exposure Function* (NEF): Esta función controla el acceso a la red por parte de aplicaciones de terceros a los que se les ofrece servicio.
- *Application Function* (AF): Aquí se enmarcan las potenciales aplicaciones que se puedan desplegar para adicionar funcionalidades a futuro en la red.
- *Network Repository Function* (NRF): Esta función se encarga de almacenar información de las funciones disponibles en la red, para así poder desplegarlas.

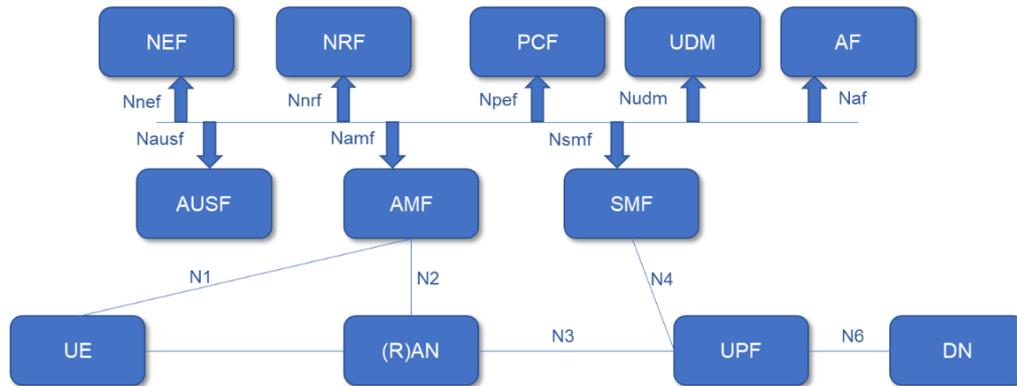


Figura 2.3: Diagrama de la arquitectura del núcleo 5G con sus diferentes funciones de red y la comunicación de ellas con el usuario o la red de acceso. También se ilustra la conexión con redes externas a través del UPF. Imagen extraída de [9].

### 2.1.2. Red de acceso

5G New Radio (5G-NR) es la evolución de E-UTRA (*Evolved UMTS Terrestrial Radio Access*), la red de acceso de LTE. Para permitir los variados casos de uso que se pretenden cubrir con 5G, NR incluye flexibilidad en la configuración de la red, algunas de estas son:

- El uso de frecuencias más altas que las utilizadas en generaciones previas, llegando hasta 50 GHz en las denominadas bandas milimétricas [33]. Esto permite mayores tasas de transferencia debido a la mayor cantidad de espectro disponible. Sin embargo, estas frecuencias tienen una menor capacidad de penetración en obstáculos así como un menor alcance debido a su mayor atenuación. Es por esto que se requieren una mayor densidad de radiobases para lograr una cobertura adecuada. El caso de uso de estas es principalmente en pequeñas áreas urbanas con una alta densidad de usuarios, como pueden ser estadios, centros comerciales, etc.
- Flexibilidad en la estructura de transmisión sobre la cual se ubican los datos, así como los mensajes de control, para ser enviados por la interfaz de radio. Esto permite adaptarla a las necesidades de los distintos casos de uso para cumplir con sus demandas específicas. Ejemplos de estos son la posibilidad de elegir el tiempo de duración del intervalo sobre el cual se asignan los recursos o la posibilidad de asignar fracciones de los recursos, de forma de ajustarse a los requerimientos de latencia más estrictos.

Si bien el despliegue que se espera en una red 5G madura es conectando 5G-NR con 5G-CN, es posible conectar la tecnología de radio de 5G con el núcleo de red LTE ya desplegado en lo que se conoce como 5G-NSA (*Non-Standalone*). Esto permite aprovechar sus mejoras sin que sea necesario desplegar todo un nuevo núcleo, con los costos que ello acarrea. Sin embargo esta configuración requiere que el usuario mantenga una conexión con un nodo LTE, ya que el tráfico que es enviado

## Capítulo 2. Marco Teórico

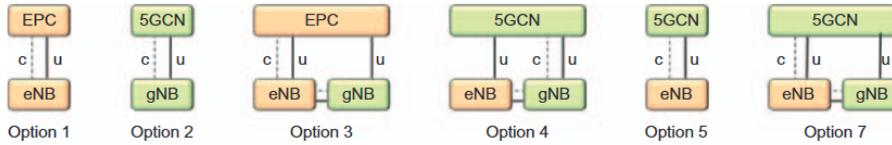


Figura 2.4: Diagrama de las diferentes opciones de despliegue de 5G y LTE en las cuales se permite flexibilidad de conexión entre el núcleo de red de una tecnología con la red de acceso de otra o ambas. Se indica la tecnología de acceso sobre la cual puede ser enviado el tráfico asociado al plano de usuario o al plano de control. Imagen extraída de [18].

por NR es únicamente el asociado al plano de usuario. Por otro lado el despliegue completo de 5G, incluyendo el núcleo de red se denomina 5G-SA (*Standalone*). En la Figura 2.4 se muestra un diagrama de las opciones de despliegue de 5G y LTE. La opción 3 corresponde al despliegue NSA, se observa que el usuario debe mantener una conexión con un nodo LTE, que es encargado del plano de control. El tráfico de datos es balanceado entre el nodo LTE y el nodo NR, aprovechando las mayores capacidades de este último.

La comunicación entre el usuario y la radiobase emplea el esquema de modulación *Orthogonal Frequency Division Multiplexing* (OFDM) debido a que simplifica la ecualización del canal permitiendo mitigar efectos indeseados en la señal transmitida. La transmisión es dividida temporalmente en símbolos OFDM, así como en frecuencia definiendo un conjunto de subportadoras. Los datos a transmitir sobre el conjunto de recursos disponibles son ubicados en una estructura de trama de más alto nivel. Se define un *slot* como 14 símbolos OFDM en el dominio temporal utilizando el prefijo cíclico básico, y 12 para el caso de prefijo cíclico extendido orientado a canales más desafiantes. Los *slots* se agrupan en *subframes* de 1 ms de duración, y 10 de estos forman un *frame*. Estos son definidos temporalmente en lugar de una cantidad de *slots* ya que estos pueden tener una duración variable. 5G-NR ofrece una nueva libertad en la elección del espaciado entre subportadoras (numerología), lo cual implica también una elección sobre la duración del símbolo OFDM. En particular, para poder dar servicios con baja latencia es de interés la utilización símbolos más cortos, sin embargo esto implica un mayor espaciado entre subportadoras y por tanto un mayor uso espectral. En las bandas de frecuencia más altas esto es de gran interés gracias a la mayor cantidad de espectro disponible.

Las distintas señales a enviar, como el tráfico asociado a los usuarios, señalización o señales de referencia, son ubicadas en función de una grilla de recursos de tiempo y frecuencia. La unidad mínima de asignación consiste de un *resource block* (RB), formado por 12 subportadoras consecutivas en el dominio de la frecuencia. En cada *slot* se define una asignación de recursos, destinando un conjunto de RB para cada transmisión requerida. La duración de un *slot*, la cual determina la frecuencia con la cual se realiza la asignación, es denominada *Time Transmission Interval* (TTI). En la Figura 2.5 se muestra la grilla de recursos de tiempo y frecuencia en 5G-NR.

La transmisión desde el usuario a la gNB es denominada *uplink* (UL) y la

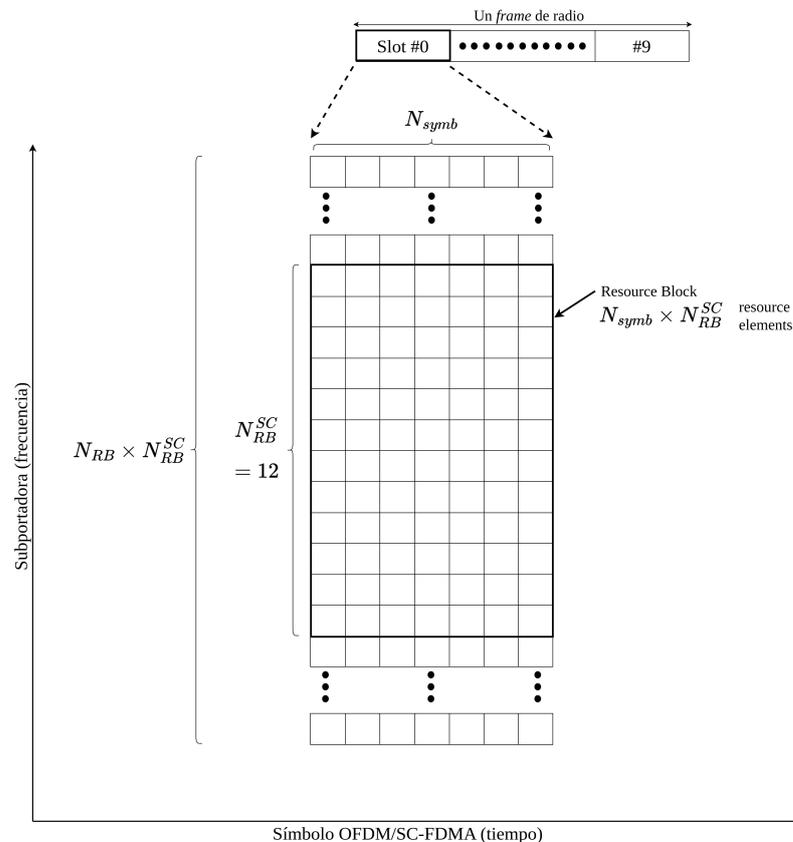


Figura 2.5: Grilla de recursos de tiempo y frecuencia en 5G-NR donde se ilustra la cantidad de subportadoras que componen un RB compuesto por la cantidad de símbolos OFDM dentro de un slot, los cuales dependen de la numerología utilizada. Imagen adaptada de [42].

transmisión en sentido inverso *downlink* (DL). Para permitir ambos es necesario definir un esquema *duplex*, lo que corresponde a una división de la transmisión para permitir traficar a cada una. Los dos esquemas más comunes son *Frequency Division Duplex* (FDD) y *Time Division Duplex* (TDD). En el primer caso se asignan bandas de frecuencia distintas, por lo que en todo momento se transmiten datos en ambos sentidos en canales separados. En el segundo caso, una misma banda de frecuencia es dividida en intervalos de tiempo, donde se multiplexan las transmisiones UL y DL. En 5G-NR se permite la utilización de ambos esquemas.

Como es usual en el contexto de redes, las funcionalidades necesarias para permitir el intercambio de datos entre el usuario y la gNB se encuentran divididas en capas, ilustradas en la Figura 2.6. Aquí se describen brevemente cada una de ellas

1. *Radio Resource Control* (RRC): Esta capa se encarga de los procedimientos del plano de control asociados a la RAN. Entre estos se encuentra el manejo la información necesaria para que un dispositivo se comunique con la red, mecanismos para avisar a un usuario inactivo cuando se tienen paquetes disponibles para enviar, funciones de movilidad, entre otros.

## Capítulo 2. Marco Teórico

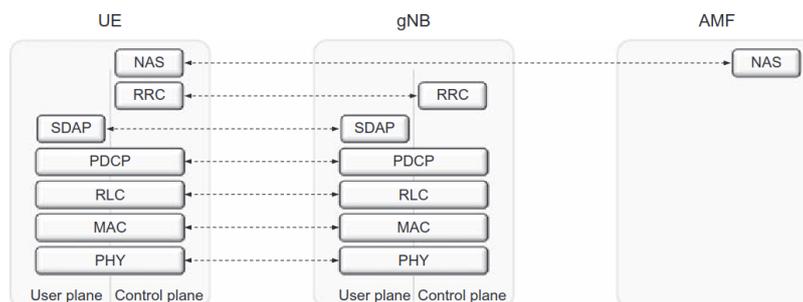


Figura 2.6: Diagrama de las capas de red involucradas en la red de acceso de 5G indicando si participan del plano de control o de usuario. Además se ilustra la comunicación del usuario con el AMF en el núcleo de la red. Imagen extraída de [18]

2. *Packet Data Convergence Protocol* (PDCP): Esta capa realiza compresión de los encabezados de los paquetes IP a transmitir utilizando un método conocido como *Robust Header Compression* (ROHC). En aplicaciones de *streaming* como pueden ser vídeos o llamadas los paquetes tienen típicamente un tamaño reducido, por lo que el tamaño de los encabezados de protocolos superiores como IP, UDP o RTP es considerable respecto al tamaño total. Por ello se aprovecha la redundancia de campos en varios de estos protocolos como el tamaño de *payload*, así como redundancia en paquetes sucesivos como direcciones IP, logrando reducir considerablemente la información a transmitir por aire. Por otra parte esta capa también se encarga del eliminado de duplicados y opcionalmente del reordenamiento de paquetes. Estas funcionalidades son de especial interés para el caso en el que un UE se traslada de un gNB a otro. Allí la capa PDCP de la gNB previa reenvía los paquetes pendientes a la nueva gNB para su entrega. En esta situación pero en dirección UL, es la capa PDCP del UE quien maneja retransmisiones necesarias, debido a que otros mecanismos de retransmisión en capas inferiores vacían su memoria al moverse de gNB [18].
3. *Radio Link Control* (RLC): Esta se encarga de almacenar en *buffers* los datos para transmitir y los envía hacia la capa MAC en los tamaños y momentos que esta solicite de acuerdo a los recursos que se tengan asignados. En recepción puede también almacenar los paquetes en *buffers* y reordenarlos de ser necesario antes de ser enviados hacia las capas superiores. Opera en los siguientes tres modos
  - *Transparent Mode* (TM): Dicho modo implica el menor nivel de procesamiento en la capa RLC, se limita a almacenar en *buffers* los datos recibidos para transmitir.
  - *Unacknowledged Mode* (UM): A diferencia del modo TM este admite la segmentación de paquetes en fragmentos mas pequeños para poder lograr ser transmitidos con los recursos disponibles. Además admite el mecanismo de reordenamiento en recepción.

- *Acknowledged Mode* (AM): Este modo realiza la mayor cantidad de funciones. Además de las correspondientes al modo UM, también implementa un mecanismo de ACK/NACK para la transmisión confiable.
4. *Medium Access Control* (MAC): En la radiobase se encarga de la asignación de recursos, determinando la cantidad de datos a enviar por cada usuario para transmitir en un TTI, en función de la cantidad de recursos disponibles. La capa MAC mantiene una única entidad para cada usuario, y por tanto es encargada de multiplexar los diferentes flujos de las capas superiores. Envía sus datos a la capa física en *Transport Blocks* (TB) con un tamaño máximo determinado por la cantidad de recursos asignados. El TB define la cantidad de RB disponibles para la transmisión. Además la capa MAC implementa el mecanismo de *Adaptive Modulation and Coding* (AMC) que permite elegir la modulación y la tasa de codificación (MCS) de los TB en función de la calidad del canal de forma de lograr una conexión estable con el usuario. Esta medida es reportada por el UE a la gNB mediante un mensaje de *Channel State Information* (CSI). En función del MCS elegido, se determina la cantidad de bits disponibles para transmitir en los recursos asignados. Además es responsable del mecanismo de retransmisiones HARQ. Este utiliza códigos para detección y corrección de errores, y toma la decisión de descartar el paquete recibido y solicitar una retransmisión en caso de que el mensaje no pueda ser recuperado correctamente.
  5. Capa física (PHY): Esta es responsable de la conversión de la señal digital (bits) a las ondas de radio que son transmitidas entre los equipos, así como la conversión en el sentido inverso en recepción. Los recursos son asignados por la capa MAC en forma de bloques de frecuencia y tiempo, y la capa física es responsable de la modulación y codificación de los datos del TB en estos recursos.

### 2.1.3. Comunicación entre el usuario y el núcleo

El tráfico de datos entre el núcleo y la red de acceso en el plano de usuario, por ejemplo paquetes IPv4 o IPv6 es encapsulado utilizando el protocolo *GPRS Tunneling Protocol* (GTP). Para cada sesión del usuario se establece una etiqueta denominada *Tunneling Endpoint Identifier* (TEID) que permite identificar el tráfico asociado. Los paquetes del túnel GTP son transportados sobre UDP/IP, sin embargo el uso de GTP facilita la gestión del tráfico entre el núcleo y la gNB como puede ser para la gestión de prioridad, así como evitar tener que manejar una gran cantidad de direcciones IP para cada usuario. En la radiobase el tráfico es desencapsulado y enviado a la capa PDCP para luego ser transmitido al usuario. En la Figura 2.7 se ilustra la comunicación entre el usuario y el núcleo de la red.

Por otra parte también es necesaria la comunicación del usuario con el núcleo para la información de control. En la red de acceso, la información es manejada por la capa RRC la cual se encuentra conectada lógicamente al usuario. Además de intercambiar información de control asociada a la RAN, la capa RRC también es

## Capítulo 2. Marco Teórico

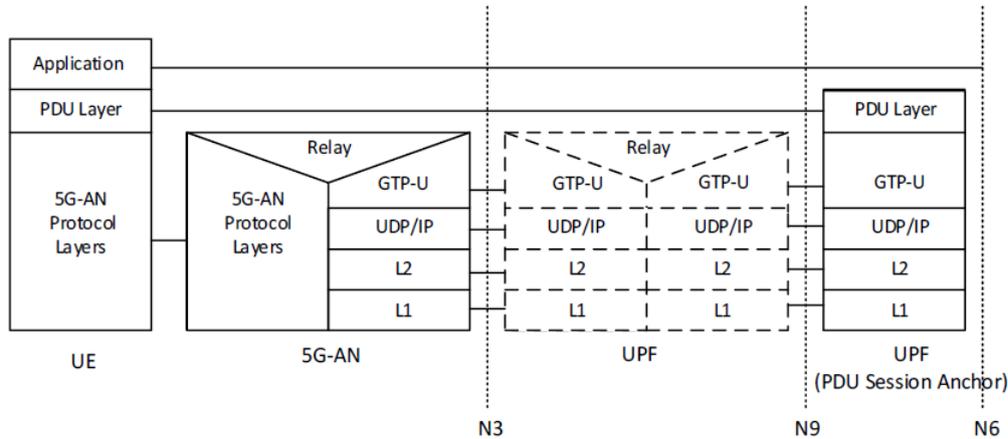


Figura 2.7: Stack de protocolos en el plano de usuario ilustrando la comunicación a nivel de cada capa entre el usuario y el núcleo de la red a través de la red de acceso. Imagen extraída de [4].

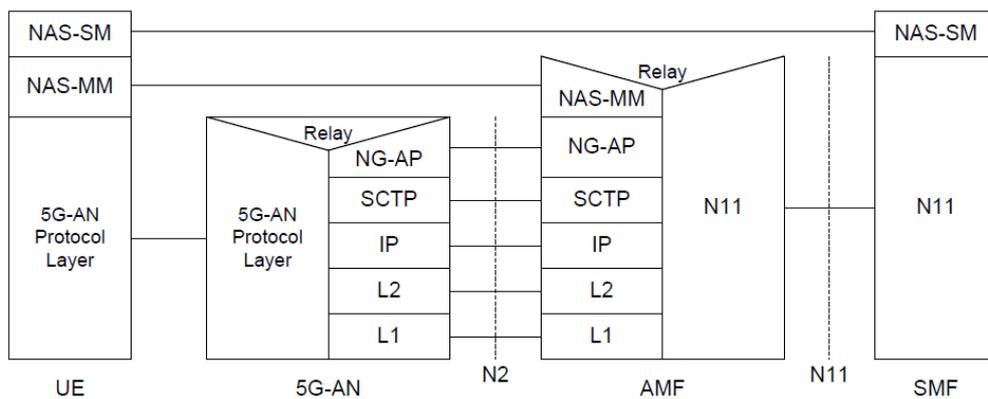


Figura 2.8: Stack de protocolos en el plano de control así como la comunicación a nivel de cada capa entre el usuario y el núcleo de la red a través de la red de acceso. Imagen extraída de [4].

encargada de encapsular el tráfico denominado *Non-Access Stratum* (NAS) que es intercambiado entre el usuario y el núcleo de la red. La entrega de los paquetes es garantizada utilizando *Stream Control Transmission Protocol* (SCTP) en la gNB. En la Figura 2.8 se ilustra la comunicación entre el usuario y el núcleo de la red en el plano de control.

### 2.1.4. Splits funcionales

Por otro lado NR también prevé cambios de la arquitectura de la RAN que permitan adaptarse a las nuevas necesidades. Ya en las redes de cuarta generación comenzó a tomar relevancia el concepto de *Centralized-RAN* (C-RAN) donde los *Baseband Unit* (BBU), responsables del procesamiento de la señal, se encuentran

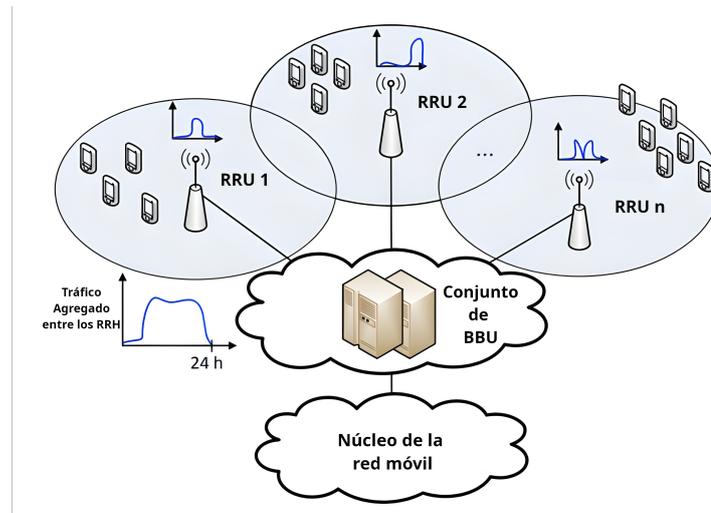


Figura 2.9: Diagrama de una red C-RAN donde se centraliza el procesamiento de la señal en un conjunto de BBU, los cuales controlan distintos RRU que se encargan únicamente de la conversión analógica-digital para transmitir y recibir las señales con el usuario final. Se muestra como el pico de tráfico puede darse en tiempos distintos para cada RRU, lo que permite un uso mas eficiente de los recursos en el BBU en lugar de tener recursos sin utilizar desplegados. Imágen adaptada de [7]

centralizados. Por otro lado los puntos de acceso consisten únicamente del *Remote Radio Unit* (RRU), también llamado *Remote Radio Head* (RRH), donde se realiza la conversión de la señal digital a radio frecuencia, comunicándose con el BBU centralizado a través del segmento de red denominado *fronthaul* como se ilustra en la Figura 2.9. Esto permite reducir los costos asociados tanto al despliegue de radiobases como a su mantenimiento [47]. Lo cual es de especial importancia al desplegar una gran cantidad para proporcionar cobertura en áreas extensas, como puede ser necesario al utilizar las nuevas bandas de frecuencia altas.

Además, centralizar el BBU (o un conjunto de ellos) permite distribuir mejor la carga y aprovechar los recursos de forma más eficiente. Los requerimientos de servicio de los usuarios poseen una gran variabilidad tanto temporalmente como espacialmente. Por ejemplo, es de esperar una mayor demanda en zonas de oficinas durante el horario laboral, mientras que en zonas residenciales la demanda será mayor en las horas de la tarde y noche [35]. En un despliegue tradicional, esto implica tener equipos con funcionalidades del BBU en cada radiobase, subutilizados en momentos de baja demanda. En un despliegue como C-RAN se puede distribuir la carga de los diferentes RRU entre los BBU de forma dinámica, permitiendo un uso más eficiente de los recursos [7]. Esto también facilita el mantenimiento y crecimiento de la red al permitir agregar nuevos sitios aprovechando la infraestructura centralizada ya existente [47].

Sin embargo, C-RAN implica transmitir la señal con prácticamente un nulo procesamiento lo cual requiere *bitrates* elevados. Esto se debe a que los datos a ser intercambiados representan las muestras temporales de la señal transmitida o recibida, lo que en particular hace que sea independiente del uso actual de la

## Capítulo 2. Marco Teórico

red y depende únicamente de parámetros del despliegue como el ancho de banda disponible. En este sentido, los datos que requieren ser enviados cuando la demanda es baja aún mantiene la misma tasa que si se encontrara sobrecargada. Con el crecimiento de los volúmenes de datos, esta demanda se verá intensificada con las mayores capacidades previstas en 5G, las exigencias sobre el *fronthaul* se ven aún más incrementadas. Debido a esto surge la necesidad de alternativas a C-RAN buscando obtener un mejor balance entre los beneficios de esta arquitectura y las tasas de transmisión necesarias para conectar los sitios. Para ello, se definen en 5G los *splits* funcionales como distintas separaciones del equipo que realiza las distintas funciones de la gNB. La 3GPP define el *Distributed Unit* (DU) como el equipo que realiza las funcionalidades en el sitio donde se encuentra la conexión con el usuario, y el *Centralized Unit* (CU) como el equipo que realiza las funciones centralizadas.

La 3GPP define 8 *splits* funcionales, ilustrados en las Figuras 2.10 y 2.11, con posibles subdivisiones para algunos de ellos. A continuación se describen brevemente las características principales de cada uno de ellos

- Opción 1: División implementada entre capas RRC y PDCP. Esta elección implica el mayor nivel de procesamiento por parte del DU dado que el plano de usuario se encuentra en su totalidad en el equipo distribuido. Esto representa la mayor complejidad de dicho equipo, lo cual es similar al de una radiobase tradicional. Debido a que casi la totalidad de las funciones se encuentran en el DU, esta opción no implica restricciones fuertes sobre la capacidad o latencia del *fronthaul*. Este es apropiado para casos de uso URLLC o *Edge Computing* donde interesa tener los recursos cerca del usuario para minimizar la latencia.
- Opción 2: Separación entre capas PDCP y RLC. Esta opción presenta características similares a las de la Opción 1 debido a que la capa PDCP no es responsable de acciones en tiempo real. Presenta la ventaja de ya encontrarse estandarizada a través de la denominada interfaz F1, ya que desde LTE existe el concepto de *Dual Connectivity*, donde un usuario se conecta a dos radiobases simultáneamente y la capa PDCP se mantiene únicamente en el nodo maestro. Además es de interés para los despliegues NSA en los cuales se requiere una coordinación entre la red de acceso 4G y 5G para el balanceo de carga.
- Opción 3: División dentro de la capa RLC. Esta corresponde a la centralización de las funciones de la parte alta de la capa RLC, entre las cuales se encuentra el mecanismo ARQ. Debido a ello se tiene una mayor sensibilidad a la latencia del enlace entre los equipos para evitar retransmisiones innecesarias. Esto posee la ventaja de poder manejar retransmisiones asociadas a distintos DU, lo cual puede ser beneficioso en casos de movilidad.
- Opción 4: Separación entre las capas RLC y MAC. Dado que en esta opción se centraliza la totalidad de la capa RLC, el tráfico sobre el *fronthaul* se ve incrementado debido al intercambio con la capa MAC para la asignación de

recursos. Esto implica que la latencia debe ser menor al tiempo de duración de un TTI de forma que los datos puedan ser enviados una vez realizada la asignación de recursos.

- Opción 5: División dentro de la capa MAC. En esta opción la parte alta de la capa MAC se encuentra centralizada, la cual incluye la asignación de recursos. Por otro lado las funciones más críticas en términos de tiempo como el mecanismo de HARQ se encuentra en el DU, por lo que no impone restricciones fuertes sobre la latencia del enlace en comparación a *splits* más bajos permitiendo distancias de más de 20 km entre los equipos [19].
- Opción 6: Entre capas MAC y PHY. Centralizar la totalidad de la capa MAC implica que las retransmisiones asociadas a HARQ se realicen en el CU. Esto implica un requerimiento de latencia estricto sobre el *fronthaul*, lo cual no permite por ejemplo grandes distancias entre los equipos distribuidos y centralizados. Este es de interés para celdas pequeñas debido a la simplificación de la unidad distribuida.
- Opción 7.3: División entre High-PHY y Low-PHY. Este *split* traslada las funcionalidades de corrección de errores al equipo centralizado lo cual puede facilitar la cooperación con la capa MAC debido a su relación por los mecanismos de retransmisión.
- Opción 7.2: División entre High-PHY y Low-PHY. En esta opción se centralizan algunas funciones más como la modulación. Su tráfico se espera que sea similar al de la opción 6.
- Opción 7.1: División entre High-PHY y Low-PHY. Aquí se da un punto de inflexión en los requerimientos del *fronthaul*, principalmente debido a que este deja de ser dependiente del tráfico para pasar a ser fijo. Esto sucede ya que se dejan de distinguir los subportadoras en las que se encuentran datos útiles y son transportados los símbolos asociados a todas ellas.
- Opción 8: División entre PHY y RF. Esta corresponde a la opción con el mayor nivel de centralización, similar a la de C-RAN. En esta opción el equipo distribuido consiste únicamente de la conversión de la señal digital a radio frecuencia, pero debiendo enviar cada muestra temporal al equipo centralizado. Esto implica un requerimiento de *bitrate* elevado en el *fronthaul*, lo cual puede ser un desafío en términos de costos y latencia. Por contraparte el equipo distribuido es de muy bajo costo, además que la centralización del procesamiento permite un manejo energético eficiente de los recursos en función de la carga del sistema [20].

En la Figura 2.12 se muestra la comparación del *bitrate* requerido en *fronthaul* en UL y DL para las diferentes opciones de *split*. Se puede observar que de acuerdo al estudio realizado en [22] las opciones de *splits* bajas de 1 a 6 mantienen un volumen de tráfico no excesivamente alto y sin mayores variaciones entre ellas. Por otro lado las opciones de *splits* de capa física, particularmente 7.1 y 8, presentan

## Capítulo 2. Marco Teórico

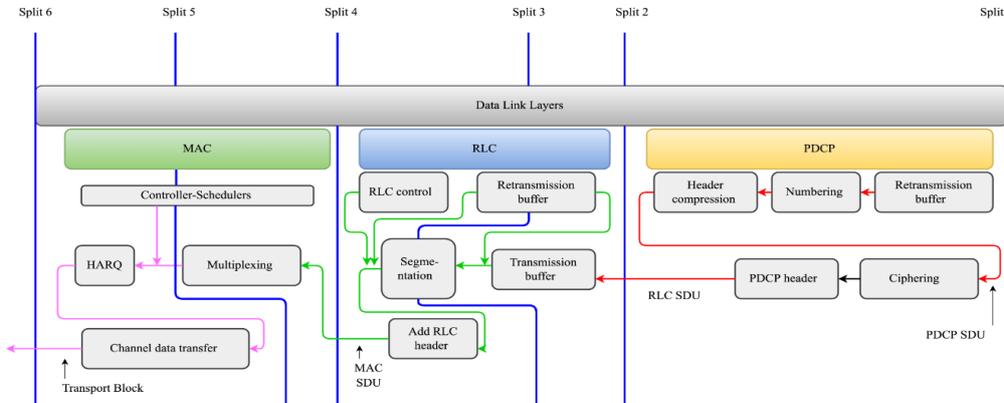


Fig. 9: Architecture of split 1 to split 6 of the data link layer: from PDCP to RLC to MAC in the downlink.

Figura 2.10: Opciones de *splits* 1 a 6, indicando las funciones presentes en cada capa así como sus interacciones y la separación concreta de estas en los *splits* que no coinciden con las fronteras. Imagen extraída de [19].

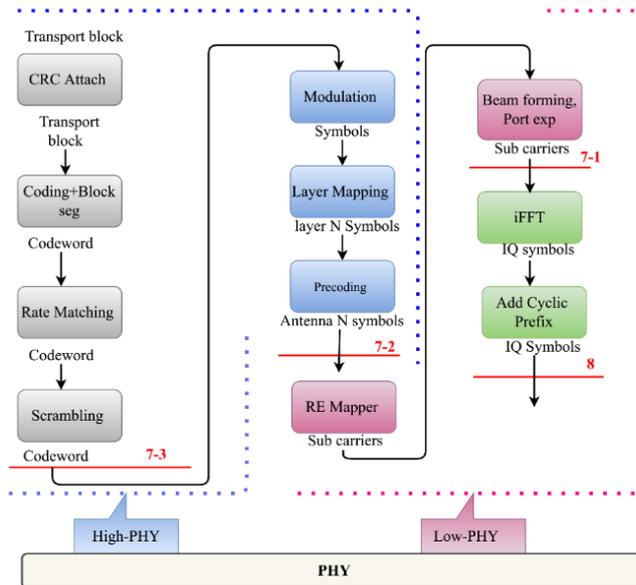


Figura 2.11: Opciones de *splits* 7 (con sus subdivisiones) y 8, indicando las funciones presentes en cada capa así como sus interacciones y la separación concreta de estas en los *splits* que no coinciden con las fronteras. Imagen extraída de [19].

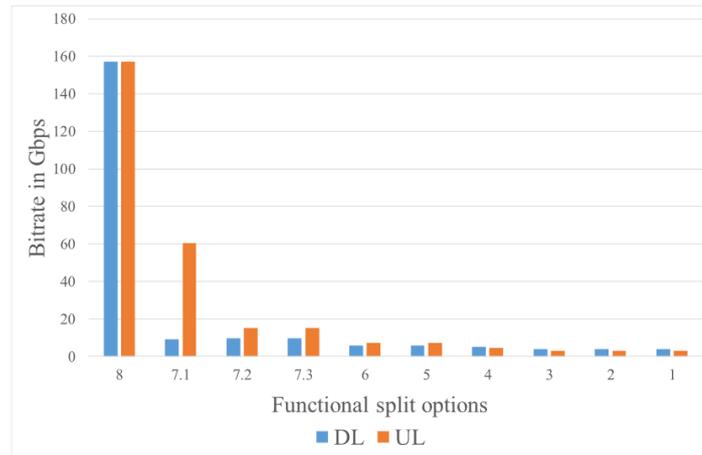


Figura 2.12: Comparación del *bitrate* requerido en *fronthaul* en UL y DL para las diferentes opciones de *split*. Imagen extraída de [22]

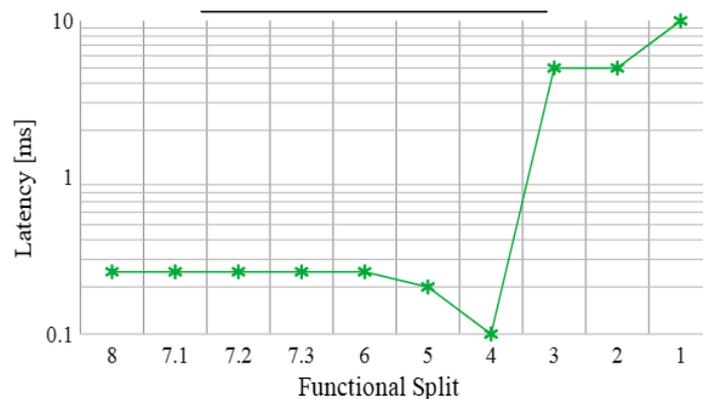


Figura 2.13: Comparación de la latencia tolerada en el *fronthaul* para las distintas opciones de *split*. Imagen extraída de [19].

un incremento considerable en el *bitrate* requerido. Lo cual es explicado por la necesidad de una tasa constante de transmisión independiente del tráfico de datos.

Además de requerir mayores *bitrates* para su transporte, los *splits* altos también presentan mayores requerimientos de latencia. En la Figura 2.13 se muestra la latencia tolerada en el *fronthaul* para las distintas opciones de *split*. Los *splits* bajos permiten manejar latencias de hasta 10 ms, por otro lado las opciones a partir del 4 requieren latencias menores a 1 ms. Este punto de inflexión se debe a funciones críticas como el mecanismo de retransmisión HARQ o la asignación de recursos que pasan a encontrarse centralizadas en el CU. Dado que estas deben operar en tiempo real para el próximo *slots* la latencia debe ser menor a la duración de un TTI, lo cual es de 1 ms como máximo dependiendo de la numerología utilizada.

Si bien en este trabajo nos centraremos en la definición de los *splits* funcionales y sus equipos dada por la 3GPP en [1], existen otras estandarizaciones bajo la misma idea. En particular se mencionan dos de ellas con un gran interés por parte

## Capítulo 2. Marco Teórico

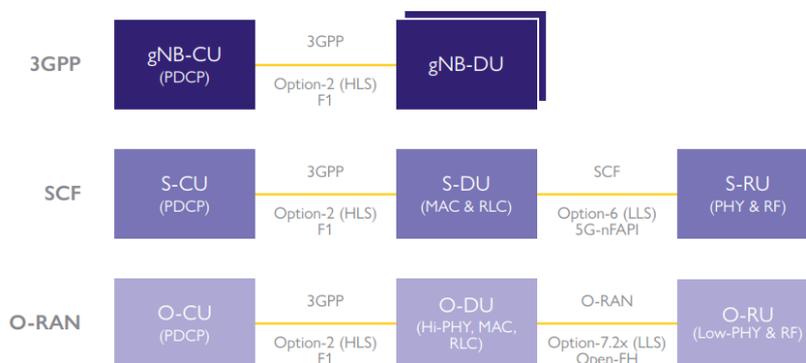


Figura 2.14: Comparación de las arquitecturas de red definidas por la 3GPP, OpenRAN y Small Cell Forum. Imagen extraída de [12].

de la comunidad y un proceso avanzado y activo.

- *OpenRAN*: El objetivo general de esta organización es la definición de interfaces abiertas para la RAN. Esto permite la interoperabilidad entre equipos de distintos fabricantes, lo cual puede ser de interés para los operadores al permitirles elegir el equipo que mejor se adapte a sus necesidades. Dado que también se tiene como objetivo la desagregación de las funciones de la red, la arquitectura de OpenRAN presenta la separación de la red de acceso en diferentes equipos. En particular se define el *Radio Unit* (RU) análogo al RRU de C-RAN, únicamente con funcionalidades de capa física. Las capas superiores se encuentran divididas en dos equipos denominados de forma similar a la 3GPP, DU y CU.
- *Small Cell Forum*: Esta organización tiene como objetivo la creación de estándares que promueven la adopción de las celdas pequeñas en las redes móviles. Entre sus trabajos se encuentra la definición de *network Functional Application Platform Interface* (nFAPI) que define la interfaces entre los equipos, con una arquitectura similar a la de OpenRAN consistente de los equipos RU, DU y CU.

En la Figura 2.14 se muestra una comparación de las arquitecturas de red definidas por la 3GPP, OpenRAN y Small Cell Forum. Dado que las dos últimas organizaciones definen la separación en tres equipos, se tiene una mayor flexibilidad en la elección de la ubicación de las funciones de la red. Los *splits* de la 3GPP pueden ser vistos como combinaciones de los definidos por OpenRAN y Small Cell Forum en las cuales el DU y el RU o el DU y CU se encuentran localizados en un mismo sitio.

Existen protocolos específicos para el transporte del tráfico asociado a la red de acceso, incluyendo el *fronthaul* y la posibilidad de implementar *splits* funcionales de 5G, a continuación se describen brevemente dos utilizados en la amplia mayoría de los casos.

- *Common Public Radio Interface* (CPRI): Este protocolo es utilizado en la comunicación entre la BBU y la RRU en las redes LTE. Este protocolo define la forma en la que se transmiten las muestras de la señal digital a través de la interfaz *fronthaul*. Si bien puede ser utilizado para los *splits* funcionales de 5G, se encuentra limitado a la opción 8 correspondiente a C-RAN.
- *enhanced CPRI* (eCPRI) [10]: Este protocolo es una evolución del CPRI, de forma de adaptarse a las necesidades de 5G. Entre ellas se encuentra la consideración de diferentes puntos de *split*, en particular las correspondientes 1, 2, 4, 6, 7.2, 7.3 y 8. Esto la hace una opción más adecuada para 5G que CPRI.

Dadas las altas tasas de transferencias que se ofrecen a los usuarios, incluso para opciones de *split* con bajo nivel de centralización, el ancho de banda requerido para la conexión entre la DU y la CU es elevado. Adicionalmente, la infraestructura debe garantizar la conectividad de los diversos sitios con el núcleo centralizado de la red, lo cual implica la agregación de tráfico proveniente de múltiples zonas. Debido a esto, la red de transporte de los operadores móviles es, en su gran mayoría, una red óptica gracias a su capacidad de cubrir grandes distancias y lograr *bitrates* altos.

## 2.2. Redes ópticas

Frente a la demanda en las redes de obtener enlaces confiables con alto ancho de banda es que se comienzan a sustituir las conexiones satélites o de cobre por fibra óptica. Ya desde los años '80, cuando se realizaron las primeras instalaciones de fibra, las ventajas sobre los otros medios eran claras. Con fibra óptica se tiene un medio inmune a la interferencia electromagnética, que no produce interferencias a equipos vecinos, y que logra transmitir información a una mayor distancia. Como varias de sus ventajas y desventajas son producto de la composición del cable, a medida que mejoran las técnicas de manufactura, la fibra óptica se vuelve un medio aún más valorado. De esta forma podemos decir que las principales diferencias con los cables de cobre son:

- Costo de instalación y reparación: Las fibras ópticas son elementos frágiles y que no admiten ser flexionados, haciendo costosa su instalación. Además, en caso de ruptura, los empalmes son costosos.
- Ancho de banda: En una instalación típica, las fibras ópticas permiten velocidades de transmisión 100 veces mayores que las de los cables. La velocidad máxima alcanzada con una fibra ya superó los Pb/s [38].
- Confiabilidad: La inmunidad frente a interferencias electromagnéticas y señales parásitas permite a las instalaciones de fibra óptica a reducir su tasa de errores incluso en ambientes con alta perturbación.

## Capítulo 2. Marco Teórico

- Distancia: Dada la baja atenuación de la señal frente a largas distancias, permite a la fibra óptica realizar conexiones transatlánticas.

En el caso de los *splits* funcionales, la baja latencia y alto ancho de banda hacen que este medio sea una buena opción para la conexión entre los CU y DU. Al mismo tiempo, ofrecen una mayor flexibilidad al permitir una gran distancia entre los nodos conectados, pudiendo tener los DU desplegados sobre zonas amplias.

### 2.2.1. Integración óptico-eléctrica

Si bien las fibras ópticas ofrecen varias ventajas, es necesario adquirir equipos especializados para poder integrarlas a las redes de telecomunicación ya existentes. Por un lado es necesario contar con módulos capaces de convertir señales eléctricas a ópticas —un Transmisor Óptico— y otro que haga lo inverso —un Receptor Óptico. Aquellos equipos que permiten una conversión bidireccional son llamados Transceptores Ópticos. Por otro lado, al tener topologías más complejas, se requieren otros equipos como repetidores o *switches*, que permiten extender la distancia de transmisión y el direccionamiento de haces de luz respectivamente. La integración de estos dominios se ve también afectada por la falta de almacenamiento óptico, perdiendo la información transmitida si esta no es procesada a tiempo.

### 2.2.2. Grilla flexible

Al momento de realizar las conexiones, se puede colocar una fibra *punto a punto* (P2P) o *punto a multipunto* (P2MP). Como en esta segunda opción varios dispositivos comparten el uso del canal, es necesario dividir los recursos disponibles eficientemente. Por un lado tenemos técnicas de *multiplexado temporal* (TDM), y por otro de *multiplexado en longitud de onda* (WDM). También existen variantes de estas técnicas como lo son *Dense Wavelength Division Multiplexing* (DWDM) o incluso una mezcla de estas dos opciones como es *Time and Wavelength Division Multiplexing* (TWDM). Cómo se puede ver en [46], [11], las distintas tecnologías afectan de distinta manera a los distintos *splits* funcionales y la elección óptima no es evidente.

En una implementación típica de DWDM, se utilizan bandas fijas de 50 GHz desde las cuales se transmite la información, dividiendo el espectro en múltiples canales con un mismo medio físico, como se ilustra en la Figura 2.15. Una gran desventaja de esta implementación es que no todas las aplicaciones requieren el mismo ancho de banda, con algunas transmisiones subutilizando su espectro, mientras que otras necesitan más de una banda. En los últimos años se han analizado las *Redes Ópticas Elásticas* (EON) y en particular las grillas flexibles, o *flexgrid*. En estas arquitecturas, se cuenta con una grilla con mayor granularidad (bandas de 12,5 GHz) que pueden ser concatenadas. Como se puede ver en el ejemplo de las imágenes 2.16 y 2.17, con esta técnica se puede asegurar una mejor utilización del espectro y permite ajustarse a las demandas de la red.

## 2.2. Redes ópticas

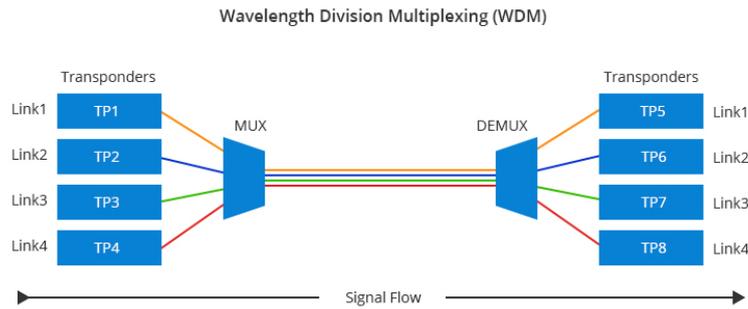


Figura 2.15: Multiplexación por división de onda en redes ópticas. Imagen extraída de [43].

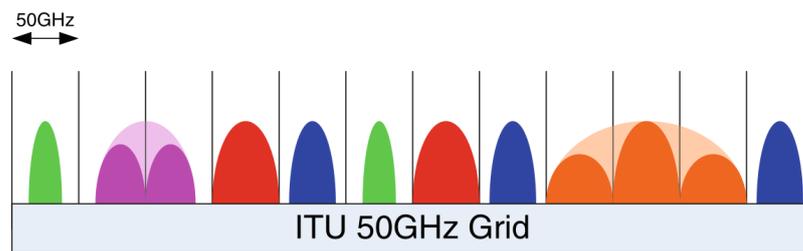


Figura 2.16: Esquema de utilización del espectro con DWDM con grilla fija. Se pueden ver transmisiones afectadas por los filtros entre bandas. Imagen extraída de [21].

En este capítulo se estudió la tecnología 5G, incluyendo los *splits* funcionales en los cuales se centra este trabajo, así como las tecnologías de redes ópticas utilizadas típicamente para el transporte entre los distintos sitios de la red móvil. En lo que sigue se presentan las herramientas de simulación y sus extensiones disponibles sobre las cuales se incorpora la simulación de los *splits*.

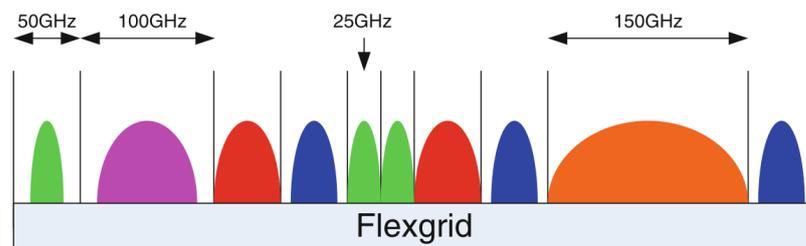


Figura 2.17: Esquema de utilización del espectro con flexgrid. La variación de los anchos de las bandas permite más flexibilidad en la transmisión. Imagen extraída de [21].

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 3

## Simuladores

En este capítulo se analizan las distintas herramientas de simulación disponibles para el modelado de redes móviles y se justifica la elección de una de ellas para el desarrollo del proyecto. Además se describen las características de la herramienta seleccionada y las extensiones ya desarrolladas que se utilizarán en el proyecto.

### 3.1. Elección del simulador

Existen una gran variedad de simuladores de redes disponibles con diferentes características. Una de las herramientas comúnmente utilizadas por su versatilidad y aplicación en otras áreas es el software de cálculo numérico MATLAB. Una de sus principales ventajas, debido a su popularidad, es la amplia diversidad de extensiones orientadas a áreas específicas. En nuestro caso con las redes móviles 5G hay dos extensiones populares en la comunidad, estas son 5G Toolbox [24] y Vienna 5G. La primera de ellas se encuentra principalmente orientada a la simulación de la capa física de la red, permite la creación de las diferentes señales y la simulación de su comportamiento en el canal. Sin embargo, su implementación precisa de bajo nivel implica un gran costo en tiempo de simulación y por ello no tiene una capacidad de escalar a redes completas. El simulador Vienna 5G, por otro lado, posee dos variantes, una de ellas es una implementación similar a la de 5G Toolbox en la que se simula de forma precisa los enlaces [37]. Por otra parte, también posee una versión que busca simular a nivel de sistema [30], implementando módulos que representan los nodos de la red como usuarios o radiobases. La principal desventaja de estas opciones está dada por el uso de MATLAB como base, el cual es un software pago lo cual lo descarta para la solución que se busca.

Uno de los simuladores más utilizados en educación e investigación es NS-3, contando con la ventaja de ser un software libre. De manera similar a los anteriores casos, se encuentran disponibles varias extensiones para aplicaciones específicas como 5G. Una de sus principales ventajas es la gran precisión de sus simulaciones, permitiendo recolectar estadísticas a bajo nivel de las simulaciones realizadas. Cuenta con una implementación de 5G llamada 5G Lena [34] la cual implementa

## Capítulo 3. Simuladores

el stack de protocolos de 5G y permite la simulación de la red de acceso. Sin embargo, la complejidad de la implementación de nuevos módulos en NS-3 es mayor que en otros simuladores, así como su curva de aprendizaje.

Otra herramienta de simulación ampliamente utilizada por la academia es OMNeT++. Para este software ya se han desarrollado varios módulos que simulan distintos componentes de una red. Gracias a su diseño modular y accesibilidad para realizar modificaciones, este programa resulta ser un buen candidato para el diseño de nuevas extensiones. Además, este cuenta con la ventaja de ser gratuito para usos académicos. Posee una implementación de redes 5G llamada Simu5G [31] la cual implementa el stack de protocolos de 5G y permite la simulación de la red de acceso.

Un aspecto importante a la hora de elegir una de estas herramientas para nuestro proyecto son los antecedentes de trabajos dentro de Facultad de Ingeniería en esta área. Teniendo esto en cuenta, los simuladores candidatos son OMNeT++ y NS-3 ya que existen otros proyectos de grado utilizándolos. Además se encuentra en desarrollo un simulador óptico en OMNeT++, lo cual resulta de interés para nuestro proyecto (para la simulación del *fronthaul*). Por otro lado, la previa existencia de módulos simulando distintos puntos de *split* funcionales, o en su defecto la facilidad con la cual se pueden programar nuevos, es un factor relevante para la elección del simulador. Sin embargo, en este sentido no se encontraron trabajos específicos asociados a los *splits* funcionales en ninguno de ellos.

En base a estos aspectos mencionados se tomó la decisión de utilizar el simulador OMNeT++. Uno de los factores de mayor peso para esta elección fue la ausencia de simuladores de la red óptica en otras plataformas para redes móviles de código abierto. Otro motivo de peso radica en la sencillez para la implementación de nuevos módulos y su diseño, gracias a que OMNeT++ podría decirse que provee un ecosistema sobre el cual desarrollar herramientas más que un simulador completo en sí [45]. Esto permite darle un desarrollo escalable al sistema, definiendo los elementos básicos y habilitando sus extensiones en un futuro. Además, al ser un simulador de alto nivel permite una mayor flexibilidad en la definición de los escenarios de simulación, permitiendo adaptarse a diversos escenarios y casos particulares.

### 3.2. OMNeT++

OMNeT++ es un simulador de eventos discretos principalmente utilizado en redes tanto cableadas como inalámbricas. Se basa en la creación de módulos los cuales pueden ser simples o ser agrupados en módulos compuestos para lograr elementos de mayor complejidad como se muestra en la Figura 3.1. Este diseño permite una gran facilidad para ampliar funcionalidades a partir de las ya provistas. Gracias a esto se encuentran disponibles paquetes que cuentan con la implementación de funcionalidades básicas para las redes como protocolos y las entidades que los utilizan.

Los módulos simples consisten en componentes con la capacidad de recibir y enviar mensajes a través de sus interfaces. Estos son los encargados de modificar

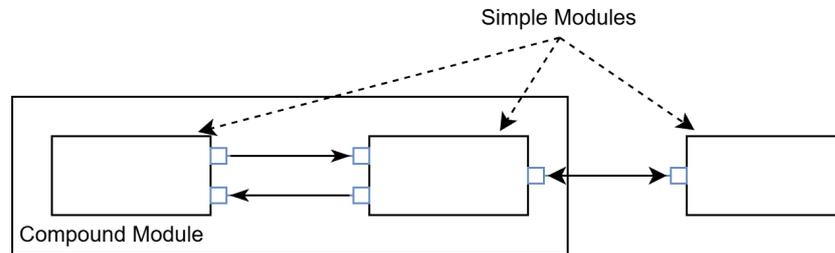


Figura 3.1: Esquema de la estructura de módulos en OMNeT++.

y operar la información contenida en los mensajes según su lógica implementada. El comportamiento se encuentra implementado en C++, lo cual permite una gran flexibilidad en la implementación de los módulos y la posibilidad de utilizar bibliotecas externas ya desarrolladas. OMNeT++ provee una librería de clases sobre la cual se deben desarrollar los módulos, estos ya cuentan con las definiciones de las funciones asociadas a lo que concierne al paradigma de simulación de eventos discretos. Se distinguen tres etapas en el ciclo de vida de un módulo durante la simulación, estas son

1. **Inicialización:** En este paso se inicializan las variables y estructuras de datos necesarias para la simulación. Esto puede incluir la lectura de parámetros de configuración u operaciones con las puertas de entrada y salida del módulo. También se puede incluir eventos iniciales que disparan el desarrollo de otros eventos como pueden ser mensajes de descubrimiento o el inicio de la comunicación con otros nodos. Esta etapa puede a su vez ser subdividida en más etapas lo que da flexibilidad para poder implementar inicializaciones en las cuales se encuentran dependencias entre distintos módulos.
2. **Desarrollo de la simulación:** Esto representa la etapa en la que se encuentra el módulo mientras se desarrolla la simulación. Aquí el módulo opera de forma reactiva a los eventos que recibe. Uno de estos eventos puede ser la llegada de un mensaje, en cuyo caso el módulo ejecuta una función ya definida por la librería OMNeT denominada `handleMessage` y cuyo comportamiento es definido en la implementación del módulo. Típicamente se verificará la puerta desde la cual se recibió el mensaje así como su tipo y se lo procesará según corresponda enviando mensajes a otros módulos, programando eventos futuros o modificando su estado. Además los módulos pueden suscribirse a señales de otros módulos, lo cual les permite reaccionar a eventos que no son mensajes enviados específicamente a ellos.
3. **Finalización:** En esta etapa se liberan los recursos utilizados por el módulo y se realizan operaciones de limpieza como por ejemplo la liberación de memoria. También puede ser utilizada para un reporte final de estadísticas de la simulación.

OMNeT++ provee una manera de recolectar estadísticas de la simulación. Para ello se definen señales sobre las cuales se pueden registrar medidas en la lógica del

## Capítulo 3. Simuladores

módulo. Posteriormente se pueden seleccionar cálculos o medidas a realizar sobre estas señales como pueden ser mínimos, máximos, promedios, histogramas, etc. Al finalizar estas pueden ser visualizadas por el usuario en la interfaz gráfica del simulador o exportadas a un archivo para su posterior análisis.

Por otra parte se define un lenguaje propio al simulador denominado *Network Description* (NED). Este lenguaje permite la descripción de módulos simples con una implementación en C++ asociada, así como la creación de módulos compuestos basados en estos módulos simples y la especificación de conexiones entre ellos. Además, el lenguaje NED permite estructurar los módulos compuestos de forma jerárquica, facilitando la escalabilidad de las simulaciones y el uso de topologías con parámetros configurables que pueden replicar topologías complejas de redes.

También se definen archivos de configuración de las simulaciones denominados INI. Estos archivos permiten la configuración de los parámetros de la simulación, como por ejemplo la cantidad de nodos, la duración de la simulación, la semilla del generador de números aleatorios, entre otros. Un mismo archivo puede contener las configuraciones asociadas a múltiples simulaciones, admitiendo que cada una de estas extienda otra definida previamente y heredando la asignación de parámetros ya realizada. Los archivos INI permiten definir asignaciones de parámetros en forma de lista o de rangos, estas generaran múltiples ejecuciones para cada configuración. Esto permite hacer estudios de parámetros para observar su impacto en el sistema. Algunos parámetros se encuentran ya definidos, como la cantidad de repeticiones de una misma simulación, lo cual permite obtener resultados en base al uso de distintas semillas para la generación de números aleatorios y obtener estadísticas más robustas.

OMNeT++ también integra un *entorno de desarrollo* (IDE) que permite una edición visual de los modelos y facilita la depuración y análisis de resultados. Además cuenta con una interfaz gráfica para la visualización de la simulación en proceso, lo que es de gran utilidad a la hora de buscar errores en la implementación, así como también para la comprensión de bibliotecas externas sin necesidad de ver su código fuente. Por otro lado, también posee una interfaz de consola dando soporte para la ejecución de simulaciones en modo *batch* para la automatización de pruebas. Esta aprovecha los recursos de núcleos de CPU disponibles para la ejecución en paralelo de las simulaciones, lo cual disminuye considerablemente el tiempo necesario para obtener los resultados.

### 3.3. INET

Uno de los paquetes más utilizados en OMNeT++ es INET. Este paquete provee una gran cantidad de módulos básicos implementados sobre la base de OMNeT++ para la simulación de redes, lo cual permite un nivel más alto de abstracción sobre las funciones básicas de OMNeT++. En particular incluye varios protocolos comúnmente utilizados en el stack de Internet. Además de permitir implementar simulaciones sencillas, establece una sólida base sobre la cual extender OMNeT++ utilizando los módulos provistos, o implementando nuevos utilizando las interfaces ya definidas. Algunas de las funcionalidades que provee son:

- Las capas de red del modelo OSI (física, enlace, red, transporte y aplicación).
- Protocolos de red como IPv4 e IPv6.
- Protocolos de transporte como TCP, UDP, SCTP.
- Interfaces de red cableadas e inalámbricas como Ethernet, 802.11, PPP, etc.
- Protocolos para enrutamiento como OSPF o BGP
- Soporte de modelado de ambientes físicos, como por ejemplo obstáculos para la propagación.

### 3.4. Simu5G

El paquete de OMNeT++ utilizado para introducir la red móvil a la simulación fue Simu5G, construido utilizando INET 4.5. Este paquete y su versión previa SimuLTE son extensiones del simulador base populares por su versatilidad para modelar escenarios sobre redes móviles de 5G y LTE respectivamente. Este provee los elementos básicos de una red celular como lo es la radio base (gNB), el móvil (UE) y el plano de usuario que permite la interconexión de la RAN y la red de paquetes. Además estos se encuentran implementados como módulos compuestos por módulos simples más específicos asociados a las diferentes capas de la red.

La biblioteca simula el plano de datos de 5G-NR tanto para la red de acceso como para el núcleo de red. La arquitectura de los principales módulos se ilustra en la Figura 3.2. En particular, los asociados a nodos de la red, usuario y radiobase, se encuentran implementados por capas de red. El usuario posee una capa de aplicación en la cual se pueden instanciar distintos tipos de casos de uso modelados mediante su generación de tráfico y lógica particular. Esto permite una gran flexibilidad y capacidad de extender el simulador para modelar distintos escenarios. Estas aplicaciones se encuentran sobre una capa de transporte, con opción de ser TCP o UDP. Ambos módulos incluyen la capa de red IP, y la radiobase incluye una interfaz punto a punto mediante la cual se conecta con el núcleo de la red. Finalmente ambos poseen una *Network Interface Card* (NIC) en la cual se implementan las funcionalidades de NR.

Se definen dos módulos auxiliares, que si bien no representan nodos de la red, son necesarios para la simulación. Estos mantienen información global de la simulación, la cual puede ser consultada por los demás nodos. El primero, y crucial para muchos aspectos de la simulación, es el módulo denominado “Binder”. Este mantiene estructuras de datos referentes a la totalidad de la red, como por ejemplo la asociación de usuarios a radiobases, la utilización de los recursos de frecuencia para un momento dado. Este manejo central de la información posee dos ventajas, por un lado facilita en gran medida la coordinación de los distintos módulos en aspectos en los cuales se influyen mutuamente. Además, permite abstraer el plano de control de la simulación, sustituyendo mensajes y servidores asociados a él por consultas o asignaciones en el Binder. Esto agiliza la creación de escenarios nuevos así como la implementación de nuevas funcionalidades. El segundo módulo auxiliar,

denominado “Carrier Aggregation” se encarga de manejar la agregación de portadoras en la red. Esta función presente en 5G permite aumentar la capacidad de la red al permitir que un usuario se conecte a varias portadoras simultáneamente.

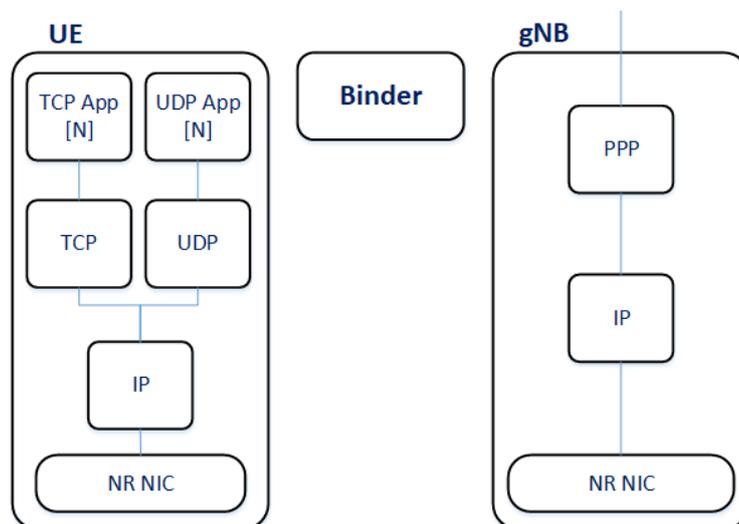


Figura 3.2: Arquitectura de los principales nodos de la red en Simu5G. Imagen extraída de [31].

La implementación del núcleo de red por parte de Simu5G se limita al UPF (User Plane Function). Este implementa la conexión de la red móvil con redes externas, típicamente Internet. La implementación actual de este módulo es simple y su comportamiento es esencialmente el de un *router*, realizando el mapeo de direcciones IP de usuarios a direcciones internas de la red 5G para su enrutamiento a las radiobases.

En particular para este trabajo es de especial interés la implementación de la NIC, ya que es allí donde se sitúan los posibles puntos de *splits*. En la Figura 3.3 se ilustra la arquitectura de la NIC tanto para UE como para la gNB. Esta sigue un diseño modular, el cual coincide con la división por protocolos del stack de 5G. La única diferencia entre la NIC de ambos módulos radica en la presencia de algunos protocolos duplicados en el usuario de forma de poder soportar una conexión tanto a la red LTE (de la versión previa del simulador) como a 5G. A continuación se describe el modelado de cada una de estas capas.

1. **Capa PDCP:** Al recibir un paquete IP, se le realiza el mecanismo de ROHC para comprimir el encabezado y se le agrega un identificador asociado a una combinación de dirección y puerto de origen y destino, este paquete es enviado a la capa inferior RLC como un PDCP *Packet Data Unit* (PDU). En el caso de recibir un paquete de la capa RLC, se descomprime el encabezado y se le quita el identificador, luego se envía a la capa superior IP.
2. **Capa RLC:** Se encuentran implementados los tres modos de transmisión TM, UM y AM. El modo TM simplemente envía los paquetes recibidos en el

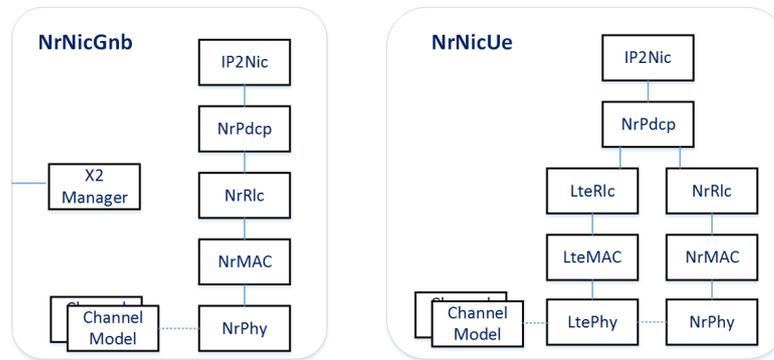


Figura 3.3: A la izquierda la arquitectura de la NIC para la gNB, a la derecha la arquitectura de la NIC correspondiente al UE. Imagen extraída de [31].

sentido correspondiente de forma transparente. En transmisión los paquetes se almacenan en *buffers* en la capa RLC hasta el tiempo de transmisión indicado por la capa MAC. En recepción los paquetes son retenidos hasta completar un PDCP PDU completo para enviarlo a la capa superior.

3. **Capa MAC:** Una de las funciones principales de la capa MAC es la asignación de recursos de radio a los distintos usuarios. Simu5G implementa la estructura de cuadro definida en 5G, permitiendo seleccionar la numerología y por tanto la duración de los *slots* (el TTI). Se permite definir la estrategia de asignación de recursos de algunas opciones ya implementadas como *Round Robin*, *Proportional fairness* y *Max C/I*. Se puede implementar un algoritmo de planificación diferente siguiendo las mismas interfaces con el planificador que las estrategias ya definidas. Además permite seleccionar el esquema DUPLEX utilizado, soportando ambas opciones FDD y TDD. Para TDD se implementan las 62 configuraciones de *slots* que define los recursos asociados a UL y DL. Se implementa un módulo con las funciones del AMC que se encarga de adaptar el MCS en base a la calidad de la señal reportada por el usuario en los *Channel Quality Indicator* (CQI). Posteriormente se elige el TB en base al procedimiento descrito en [2]. También se implementa el mecanismo de HARQ para retransmitir paquetes en caso de error.
4. **Capa PHY:** La implementación de la capa física no busca simular la conversión de las señales digitales a analógicas ni su comportamiento específico en el aire, si no más bien su efecto sobre la transmisión de los paquetes, en particular su probabilidad de error. En la Figura 3.4 se ilustra el proceso mediante el cual se determina si un paquete fue correctamente recibido o no. En primer lugar se parte de la potencia transmitida por el origen y se aplica un modelo canal ya implementado para determinar la potencia recibida. Luego se calcula la relación señal/interferencia a ruido (SINR) para cada bloque de recurso, en base a un modelo de ruido gaussiano y consultando al Binder por otros módulos transmitiendo en el mismo bloque de recurso.

## Capítulo 3. Simuladores

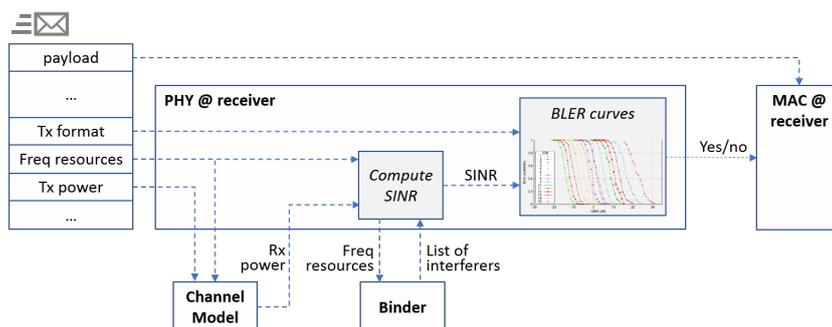


Figura 3.4: Proceso de recepción de un paquete en la capa PHY de Simu5G. Imagen extraída de [31].

Finalmente se determina si el paquete fue correctamente recibido en base a curvas de probabilidad de error en función del SINR.

### 3.5. Optical Network Simulator

En el marco del proyecto de CSIC: ‘Convergencia entre redes 5G/6G y redes ópticas: un enfoque holístico’ y con aportes del Proyecto de Grado: ‘Herramienta de simulación de una red *end-to-end* óptico-móvil’ [40] [41], está en desarrollo un paquete de OMNeT++ que ofrece varios componentes para simular una red de fibra óptica. También, se cuentan con módulos que permiten la integración de la red óptica con la red móvil. Se verá una breve descripción de los módulos implementados en esta biblioteca.

- **OpticalFiber:** Canal de fibra óptica. Simula atenuación por distancia así como también calcula la dispersión cromática y efecto Kerr. Si bien está implementado un retardo de propagación, se utiliza un valor constante independiente del largo de la fibra. Sus parámetros son: longitud de la fibra, atenuación, dispersión cromática y efecto Kerr.
- **OpticalSwitch:** Realiza la conmutación de las señales ópticas en base a una tabla de enrutamiento la cual contiene para cada puerto de entrada y longitud de onda el puerto de salida correspondiente. Su único parámetro es esta tabla la cual debe ser indicada en la configuración de la simulación.
- **OpticalTransmitter:** Representa el transmisor óptico encargado de convertir la señal digital en haces de luz. Se debe configurar la longitud de onda y la potencia de transmisión, para esta última se cuentan con parámetros para su máximo y mínimo así como su variación en cada transmisión. Además posee un retardo de procesamiento fijo.
- **OpticalReceiver:** Modela el sensor óptico en el receptor encargado de digitalizar nuevamente la señal óptica, contiene un parámetro de sensibilidad en

### 3.5. Optical Network Simulator

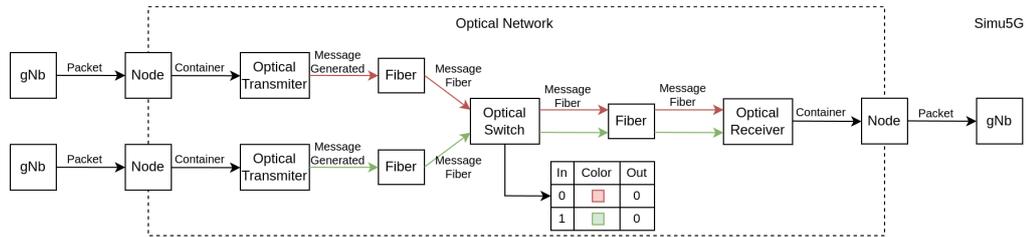


Figura 3.5: Diagrama de una comunicación entre nodos de la red móvil a través del simulador óptico. El módulo “Nodo” se encarga de traducir los mensajes entre ambos dominios. Se ilustra la transmisión desde dos gNB con un camino individual asociado a cada una hasta un *switch* óptico utilizando longitudes de onda diferentes (representadas con colores distintos). Este combina ambas señales y las envía por un único canal de fibra hacia la gNB destino.

base al cual se decide si el mensaje recibido tiene la potencia suficiente para ser interpretada. También posee configurado un retardo de procesamiento fijo.

- **Node:** Nodo traductor bidireccional entre el dominio óptico y móvil. Este módulo crea los mensajes con los parámetros necesarios para cada segmento además de la carga útil, permitiendo la integración de módulos ópticos y móviles dentro de una misma simulación.

Para la comunicación entre estos nodos se definen además el siguiente conjunto de mensajes:

- **Container:** Utilizado para encapsular los paquetes de la red móvil por el modulo “Nodo” de forma que los nodos ópticos puedan transmitirlos.
- **MessageGenerated:** Dicho mensaje es generado por el transmisor óptico para representar la conversión del mensaje a un haz de luz, para ello contiene parámetros de potencia y longitud de onda según la configuración realizada.
- **MessageFiber:** Es generado a la salida del modulo asociado a la fibra óptica, además de los parámetros del mensaje anterior también incluye los asociados a la dispersión cromática y el efecto Kerr en base a los cuales se simula el impacto de la fibra óptica sobre la transmisión.

En la Figura 3.5 se ilustra un ejemplo de la utilización de estos módulos para implementar una comunicación óptica entre tres nodos de la red móvil. Se ejemplifica el uso del *switch* óptico para combinar dos señales distintas en una misma fibra, haciendo una multiplexación por longitud de onda. En lo que refiere a este trabajo, se utilizará de igual manera el módulo “Nodo” para poder enviar los mensajes entre los equipos DU y CU a través del *fronthaul* óptico.

Este simulador también incluye los elementos correspondientes a una red *flexi-grid* en la cual se admite la asignación de canales de ancho de banda variable. Sin embargo, en este trabajo se utilizará únicamente la funcionalidad de la red óptica de forma básica, en la cual se asume una única fibra dedicada para la transmisión de los equipos DU y CU.

### Capítulo 3. Simuladores

Habiendo presentado tanto el simulador sobre el cual se basa este trabajo, así como las extensiones implementadas por la comunidad que aportan los elementos de la red móvil y óptica, en el siguiente capítulo se describe el desarrollo realizado para introducir los puntos de *split* en la herramienta base justificando las decisiones tomadas.

# Capítulo 4

## Implementación

En este capítulo se detalla la implementación de los puntos de *split* en Simu5G. Se describen las opciones de diseño consideradas así como una justificación de las decisiones tomadas. Se detallan los módulos implementados y sus funcionalidades, además de los parámetros configurables.

### 4.1. Biblioteca Simusplit

Se creó una nueva biblioteca basada en Simu5G versión 1.2.2, llamada Simusplit. Dentro de esta biblioteca se encuentran todos los módulos necesarios para ejecutar las simulaciones de los puntos de *split*. Inicialmente, se consideró incluir únicamente los módulos modificados y agregados, pero debido a que su funcionamiento depende en gran medida de otros módulos de Simu5G, se decidió hacer una copia completa de la biblioteca. En un futuro, este trabajo podría ser integrado en la biblioteca Simu5G, incluyendo las modificaciones necesarias al código ya existente así como los nuevos módulos. En su defecto, el código de este trabajo contiene una copia de la versión actual de la biblioteca lo que permitirá utilizarla de forma independiente, aún si se producen cambios en Simu5G. Al momento de definir la biblioteca, se tuvo en consideración que no haya superposición entre los nombres de los módulos y los tipos de mensajes en Simusplit y Simu5G. Esto garantiza que se puedan incluir módulos de ambas bibliotecas en una misma simulación sin conflictos. Agregando un sufijo ‘\_split’ a todos los módulos y mensajes —tanto los que están incluidos en Simu5G como los que fueron creados a lo largo del proyecto— se puede asegurar que no habrá superposición.

### 4.2. Split Manager

Una solución que se consideró para implementar los *splits* funcionales fue crear nuevos módulos correspondientes a cada opción para los equipos DU y CU, modificando el comportamiento de los submódulos asociados a cada capa para conseguir que los mensajes sean enviados a donde corresponde según el punto de *split* buscado. Dado que en los módulos implementados por Simu5G se verifica que los

## Capítulo 4. Implementación

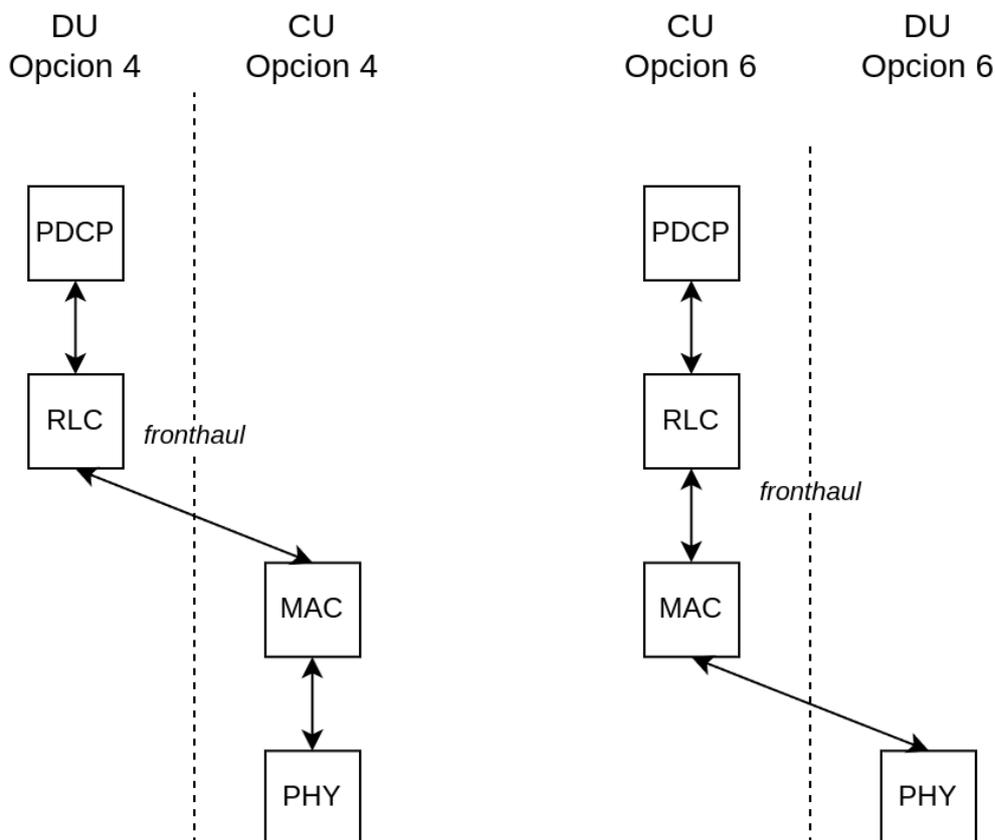


Figura 4.1: Implementación de los *splits* en base a modificar el comportamiento de cada capa en función del punto de *split* configurado. Se ejemplifica con las DU y CU asociadas a las opciones 4 y 6.

módulos conectados sean los correctos, para crear módulos específicos asociados a cada punto de *split* también es necesario duplicar y modificar todos aquellos que realizan estas verificaciones. Por ello, con esta estrategia surge el problema que para cada punto de *split* distinto que se quiera implementar, es necesario duplicar y modificar una gran cantidad de módulos (más aún si se necesitan crear módulos distintos para la DU y CU). Además, dado que esta solución implica módulos distintos para cada punto de *split* para las DU y CU, como se observa en la Figura 4.1, realizar un cambio en el punto de *split* implica un cambio de la topología de la red. Como es necesario definir las topologías previo a la simulación, realizar un cambio de *split* durante la simulación no es posible. En el Capítulo 1 se mencionó que una de las líneas de investigación con mayor interés en la comunidad es la implementación de algoritmos para la decisión dinámica del punto de *split* en tiempo real y es un objetivo de este trabajo el permitir soportar esta opción a futuro. Debido a estos motivos se decidió no seguir este diseño para la implementación.

Con el objetivo de minimizar la cantidad de archivos a modificar, se buscó una solución en la cual el módulo para los equipos DU y CU sea el mismo. Del mismo modo, se buscó un método que permita modificar fácilmente el punto de

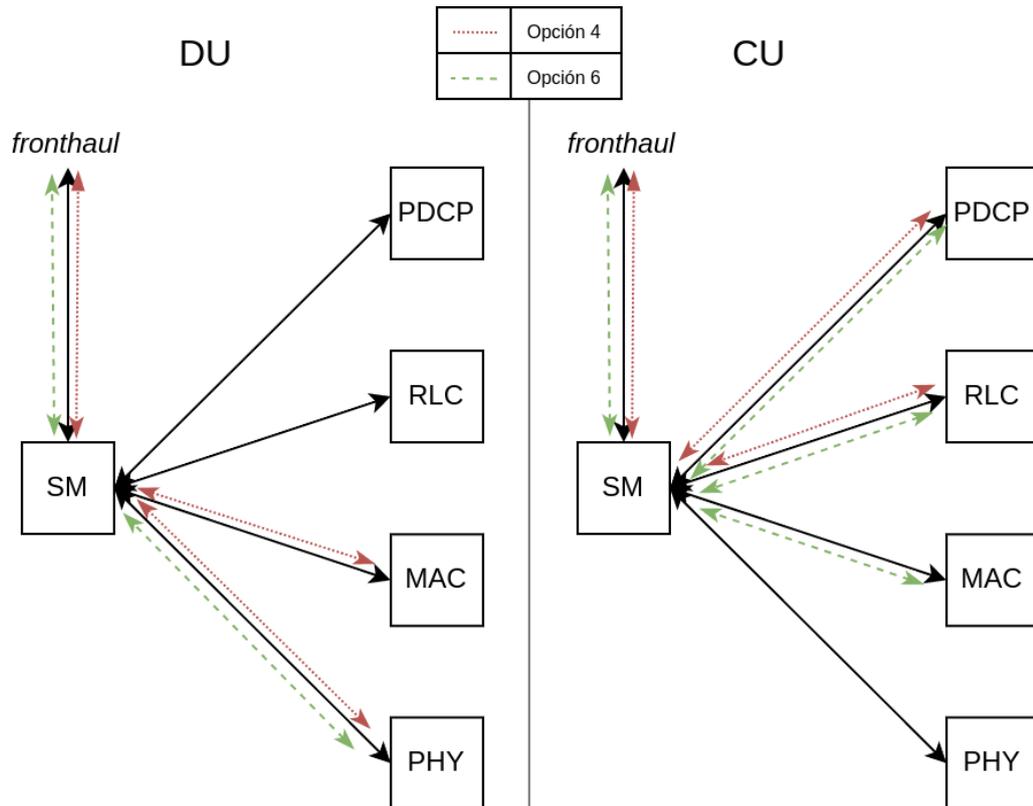


Figura 4.2: Implementación de los *splits* en base a un módulo centralizado, el Split Manager (SM), que se encarga de redirigir los paquetes entre las diferentes capas. Se ejemplifican los enlaces utilizados en la CU y DU para las opciones 4 (rojo) y 6 (verde).

*split* implementado, inclusive durante una simulación. Para ello, se creó un nuevo módulo —denominado Split Manager— como controlador central de los *splits* y encargado de manejar el flujo de paquetes dentro de la NIC de las gNB y eNB. En la Figura 4.2 se puede observar un diagrama de esta opción, en la cual la arquitectura es la misma tanto para las DU como para las CU así como también es independiente del punto de *split* configurado. En este funcionamiento de la NIC, cada submódulo envía todo su tráfico al Split Manager, y es este quien decide en función del punto de *split* configurado y del sentido del tráfico (UL o DL), a dónde reenviar los mensajes entrantes. El destino de esta decisión puede ser otro submódulo dentro de la NIC del mismo equipo —en caso de no tratarse de la frontera del punto de *split* utilizado— o es enviado a través de la interfaz del *fronthaul* al otro extremo de la conexión CU-DU —si el equipo actual no realiza la siguiente función de red que corresponde—. En la Figura 4.3 se muestra la vista de la NIC asociada a uno de los equipos en la interfaz visual de OMNeT++ con el módulo Split Manager conectado a cada una de las capas de red. Las conexiones entre capas aún deben mantenerse debido a las verificaciones realizadas sobre los módulos conectados en sus implementaciones por parte de Simu5G.

El módulo Split Manager cuenta con una interfaz de entrada y otra de salida

## Capítulo 4. Implementación

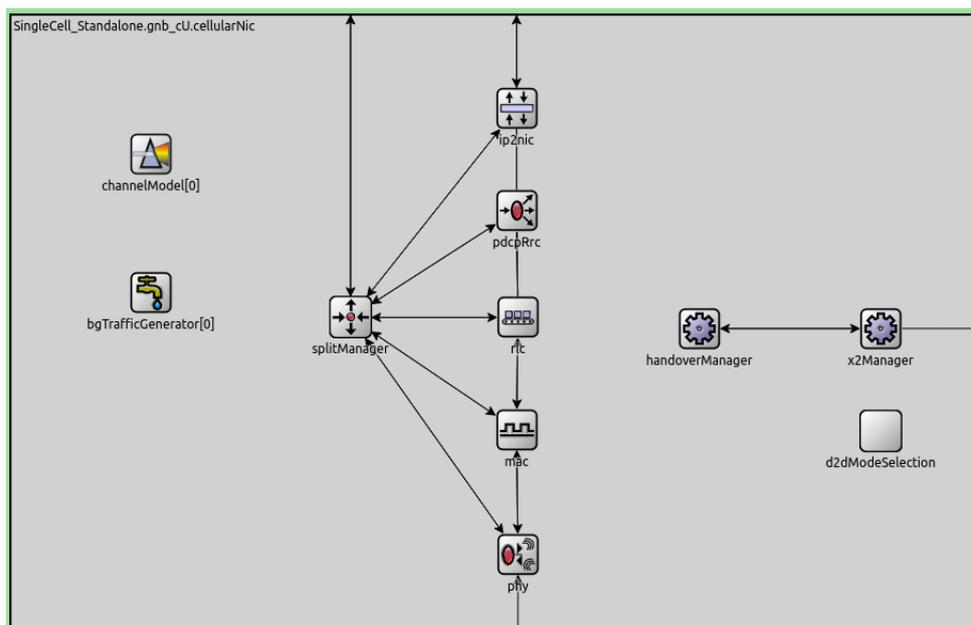


Figura 4.3: Diagrama de bloques de los submódulos de la NIC de una CU en una simulación de OMNeT++. Se puede ver al módulo Split Manager conectado con los módulos IP2NIC, PDCP, RLC, MAC y PHY, al igual que una conexión externa con una antena DU.

para cada uno de los submódulos de la NIC: IP2NIC, PDCP-RRC, RLC, MAC y PHY. Además, cuenta con las puertas SPLIT\_in y SPLIT\_out que permiten la comunicación con uno o más Split Managers a través del *fronthaul*. En OMNeT++ se señalan con paréntesis rectos la posibilidad de que haya múltiples conexiones a una misma interfaz. En concreto, las interfaces con las cuales cuenta el módulo son:

- IP2NIC\_in
- IP2NIC\_out
- PDCP\_in
- PDCP\_out
- RLC\_in
- RLC\_out
- MAC\_in
- MAC\_out
- PHY\_in
- PHY\_out
- SPLIT\_in[]
- SPLIT\_out[]

En la Figura 4.4 se muestra el funcionamiento de la NIC cuando no se configura ningún punto de *split*. Se puede ver en el diagrama que en las conexiones desde y hacia el Split Manager no hay retardo temporal. De esta manera, la presencia del módulo no afecta el comportamiento temporal del sistema. El diagrama de la Figura 4.5 muestra que cuando se implementa un punto de *split*, como puede ser el 2, el Split Manager es el encargado de la comunicación entre la CU y la DU.

## 4.2. Split Manager

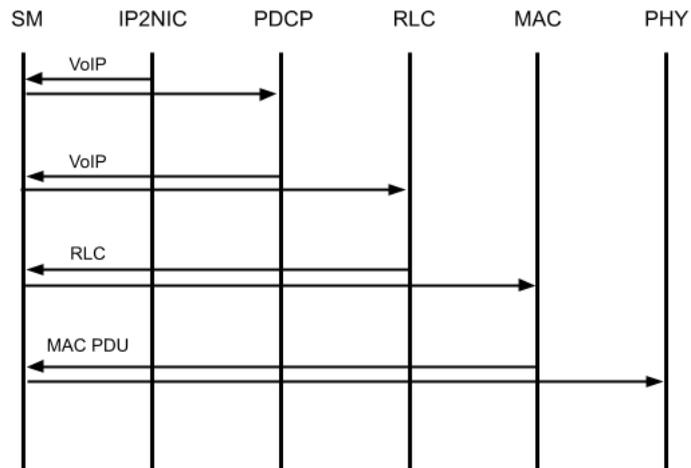


Figura 4.4: Diagrama de tiempos del funcionamiento de la NIC de una gNB sin implementar ningún punto de *split*, para un flujo DL. Los mensajes ingresan por el módulo IP2NIC y salen por el PHY. En este diagrama el módulo Split Manager está referido como SM.

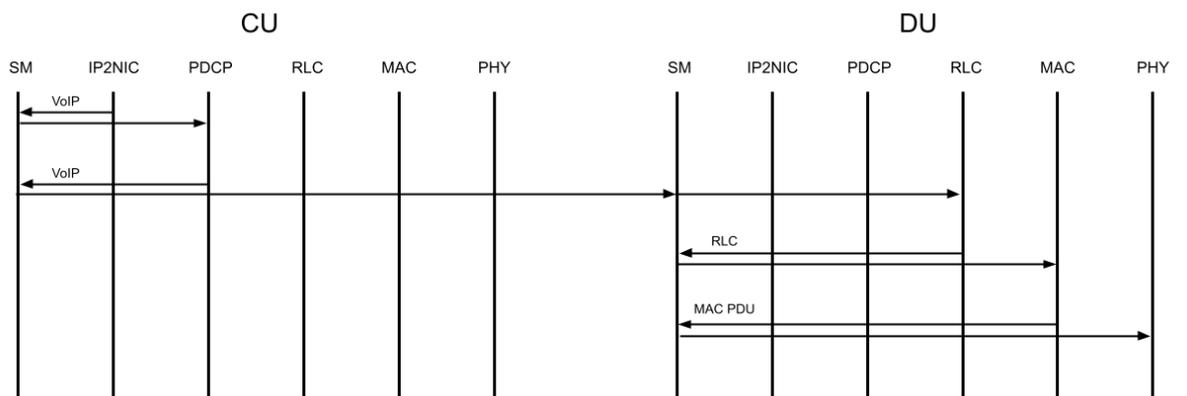


Figura 4.5: Diagrama de tiempos del funcionamiento de la NIC de una CU y DU conectadas, funcionando en el punto de *split 2*, para un flujo DL. Los mensajes ingresan por el módulo IP2NIC de la CU y salen por el PHY de la DU. En este diagrama el módulo Split Manager está referido como SM.

## Capítulo 4. Implementación

### 4.2.1. Conexión de múltiples DU a una CU

En las especificaciones de la 3GPP [1], se incluye la posibilidad de que se conecte una misma CU a más de una DU. Por lo que además buscamos que el módulo correspondiente a la CU pueda controlar y coordinar a múltiples DU conectadas de forma conjunta. Para ello, las interfaces del módulo Split Manager permiten la conexión de múltiples DU a una misma CU. Esto se debe a que las interfaces que comunican a las DU con las CU son vectores de largo variables, permitiendo agregar múltiples conexiones.

En dirección UL esta implementación es suficiente, ya que todos los paquetes son enviados a la CU y luego enrutados hacia el núcleo de la red. Sin embargo, en dirección DL, la CU debe ser capaz de identificar a qué DU corresponde cada paquete recibido y enviarlo por la puerta de salida correspondiente. Para ello se implementó una tabla de direcciones en la CU, la cual posee una asignación entre gNB destino y puerta de salida correspondiente, de forma que inspeccionando el paquete se pueda determinar cómo enviarlo a la DU correspondiente.

### 4.2.2. Capacidad de procesamiento

Si buscamos evaluar y comparar el desempeño de distintos puntos de *split*, es necesario modelar los retrasos introducidos en cada nodo DU y CU de la topología. En primer lugar se implementó un parámetro configurable —*processingDelay*— asociado al tiempo de procesamiento de cada nodo. Este modela el tiempo que se demora en analizar el paquete en el nodo antes de enviar la respuesta, afectando a los paquetes entrantes por la interfaz del *split*. Durante el transcurso de este tiempo de procesamiento, el Split Manager no continúa el procesamiento del siguiente paquete en la cola de espera. Este parámetro puede ser asignado con un valor constante, o con una distribución de probabilidad sobre la cual se sorteará el tiempo de procesamiento de cada paquete. Estas pueden ser elegidas de un conjunto de distribuciones ya implementadas por OMNeT++ como exponencial, normal, Erlang, etc.

Por otro lado, también se modeló la sobrecarga de un nodo, implementando una cola de espera de tamaño limitado para los paquetes entrantes. Para ello se utilizó el módulo `PacketQueue` de la biblioteca INET. Este módulo permite definir, además de su tamaño, el criterio con el cual se descartarán los paquetes, así como un criterio de ordenamiento. Este último permite implementar distintas políticas de prioridad sobre los paquetes en la cola. Con el fin de permitir a un equipo CU manejar un conjunto de varios DU, se implementó una cola de espera distinta para cada uno de los DU. Estas son procesadas por el mismo nodo, y la planificación sobre cada una es realizada por un módulo `PriorityScheduler` también de la biblioteca INET. La decisión implementada por este consiste en retirar de la primera que posea paquetes en espera del conjunto de colas de espera conectadas. Este comportamiento puede ser modificado en un futuro para adaptarse a otros escenarios. La biblioteca INET posee otros planificadores ya implementados que pueden sustituir al elegido y además permite la creación de uno propio. Esto es posible ya que todos estos deben extender una clase base la cual define sus in-

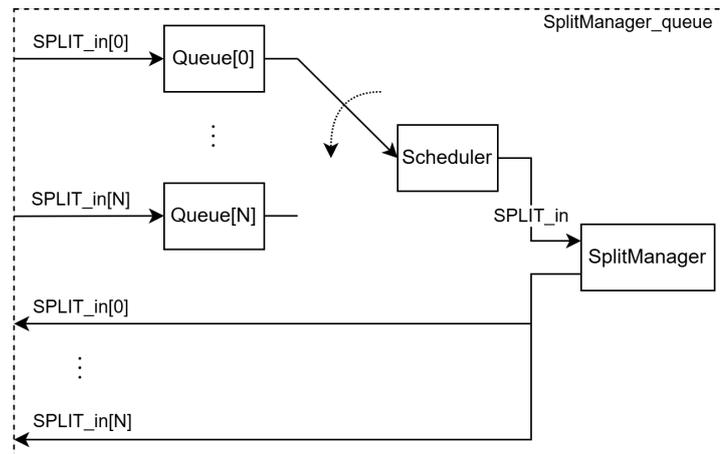


Figura 4.6: Diagrama del módulo `SplitManager_queue` con la capacidad de conectar  $N$  equipos en la interfaz del *split*.

terfaces, lo cual las hace intercambiables sin modificar el comportamiento de los módulos conectados. En la Figura 4.6 se puede observar la estructura del módulo Split Manager incluyendo el modelado de los recursos de procesamiento.

### 4.2.3. Opciones de *split* implementadas

Como fue mencionado previamente, Simu5G posee implementadas las capas del stack de protocolos de acceso como módulos separados. Esto facilita la implementación de aquellas opciones de *split* que coinciden con los límites de estas capas, como son el 2, 4 y 6. Debido a esto se tomó la decisión de limitar la implementación a estas opciones, ya que la división de funcionalidades asociadas a una misma capa implicaría una reestructuración de los módulos ya implementados. Esto puede ser complejo debido al entrelazamiento de las funcionalidades de cada capa por decisiones de diseño de los autores. Por otra parte, se planteó el objetivo de la implementación un *split* de capa física como son las asociadas a la opción 7 debido al gran interés de la comunidad en este tipo de *splits*. Sin embargo, dado que OMNeT++ se encuentra orientado a ser un simulador de sistemas completos y por tanto de mayor alto nivel, la abstracción de la capa física implementada por Simu5G no es suficiente para poder implementar dicho *split*. Para ello se requeriría un modelado de la cadena de procesamiento de la capa física como lo es el mapeo a símbolos, la modulación de OFDM o la conversión analógica-digital. En particular es de interés el modelado de la tasa de bits generados luego de cada una de estas funciones, ya que es uno de los principales impactos en el sistema al utilizar estas opciones de *split* como se mencionó en el capítulo 2.

Las opciones de *split* 2 y 4 admiten topologías en las cuales se conecta una CU a varias DU. El motivo por el cual la opción 6 no lo implementa es debido a la necesidad de manejar la asignación de recursos de forma centralizada. Esto presenta desafíos debido a la implementación de Simu5G, así como en el diseño de este en un sistema real. Una opción sería instanciar una capa MAC para cada

## Capítulo 4. Implementación

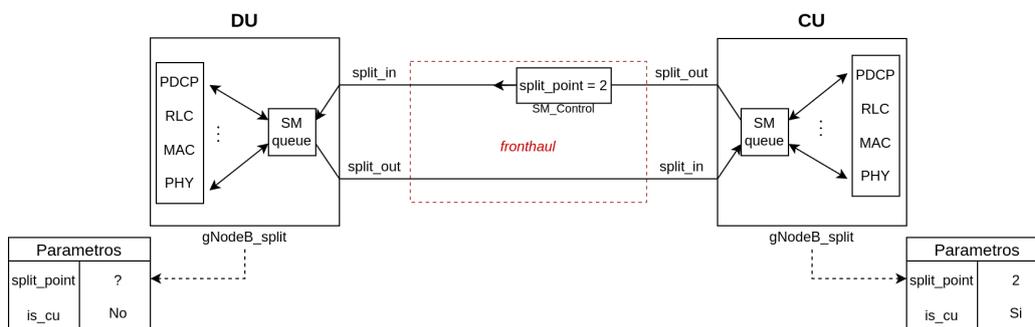


Figura 4.7: Diagrama del módulo `gNodeB_split` implementado y sus interfaces externas, con la capacidad de representar tanto a los equipos DU como a los CU dependiendo de sus parámetros configurados. Se ilustra que la DU puede no tener un *split* configurado inicialmente y cómo la CU le configura su propia opción mediante un mensaje de control.

equipo DU conectado y que estos funcionen de manera independiente. Otra opción, más interesante a la vez que más compleja, es mantener un control centralizado de todos los equipos a la vez lo cual permitiría una coordinación de estos, que puede ser beneficioso para considerar escenarios con celdas no muy distanciadas donde pueden interferirse entre si.

### 4.3. Módulo DU/CU

La biblioteca implementada posee un único módulo nuevo a incluir en las topologías de la red móvil a alto nivel, denominado `gNodeB_split`. Este sustituye al módulo `gNodeB` de Simu5G, y representa tanto a los equipos DU como a los CU distinguiendo entre ambos en base a un parámetro que puede ser establecido en los archivos INI de las simulaciones. Cada uno de estos módulos posee una NIC como la que se ilustró en la Figura 4.3, incluyendo el módulo `SplitManager_queue` con la capacidad de conectar varios equipos a la interfaz del *fronthaul* y las colas de paquetes asociadas a cada uno, siguiendo el diagrama de la Figura 4.6. En la configuración de la simulación además de identificar cada nodo como DU o CU, se cuenta en cada uno con un parámetro para indicar el punto de *split* a utilizar. Si bien este puede ser configurado tanto para el equipo centralizado como para el distribuido, al iniciar la simulación el equipo CU configurará a los DU que disponga conectados con su elección propia a través de un mensaje, de esta forma se mantiene la coherencia de la topología. Este mensaje realiza un cambio sobre el *split* configurado en los equipos DU en tiempo de ejecución, lo que cumple con el objetivo planteado acerca de permitir una asignación dinámica de la opción para trabajo futuro. En la Figura 4.7 se muestra el nuevo módulo disponible, instanciado como una DU y una CU así como el comportamiento mediante el cual se configura el punto de *split* al iniciar la simulación.

### 4.3.1. Mensajes SMControl

Con el objetivo de permitir el intercambio de información acerca del *split* entre los equipos, se definió un nuevo tipo de mensaje de OMNeT++ denominado SMControl asociado a información de control del Split Manager. Estos son intercambiados e interpretados entre los módulos Split Manager de las CU y DU. Mediante su uso es posible configurar de forma dinámica los parámetros del controlador de *split*. Se implementaron dos tipos de mensajes:

- SplitSet: Enviado por la CU al DU para configurar en ella el punto de *split* a utilizar. Se configuró que al inicializar la simulación, la CU envía un mensaje a todos los DU conectados enviándoles su `cu_name` y (con el parámetro `split_point`) en qué punto de *split* deben configurarse.
- MacNodeId: Enviado por la DU al CU al comienzo de la simulación para que la CU pueda asociar dicha interfaz con el identificador de la DU correspondiente en la tabla anteriormente mencionada y así poder enviar paquetes a través de la interfaz del *fronthaul* al destino correcto en DL.

### 4.3.2. Parámetros

Al momento de añadir las DU y CU a las topologías, es necesario configurar en el archivo INI los siguientes parámetros:

Parámetro	Tipo de dato	Uso
<code>split_point</code>	Entero	Punto de split a utilizar
<code>is_cu</code>	Booleano	Indica si la antena corresponde a un CU o un DU.
<code>num_split_connections</code>	Entero	Cantidad de DU conectados al mismo CU. Solo configurado para el CU.
<code>processingDelay</code>	Double	Valor para el <i>delay</i> de procesamiento. Admite una distribución de probabilidad
<code>queueSize</code>	Entero	Tamaño de la cola de espera.

El parámetro `num_split_connections` debe ser configurado de acuerdo a la topología a implementar. Como se explicó anteriormente, se buscó que los módulos NIC de las DU y CU sean iguales, por lo que el parámetro `is_cu` determina cómo funciona el modulo. Se cuenta con el parámetro `split_point`, que indica el punto de split en el cual debe operar el equipo y es implementado de forma que pueda variar durante la simulación. Por último también se encuentran los parámetros `processingDelay` y `queueSize` para el modelado del tiempo y capacidad de procesamiento respectivamente.

En el Apéndice A se encuentran algunos cambios menores que fueron necesarios realizar en otros módulos de Simu5G con el fin de que la simulación se siguiera realizando de manera correcta. A modo de ejemplo, algunos de estos refieren a las direcciones a utilizar entre el DU o CU para enrutar los paquetes.

Habiendo diseñado el módulo Split Manager y teniendo implementados 3 puntos de *split* (2, 4 y 6), se construyen distintos experimentos para evaluar el desempeño de esta herramienta.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 5

## Experimentación

En este capítulo se presentan las topologías utilizadas para la validación de la herramienta desarrollada. Además se especifican los parámetros fijos de las simulaciones así como los experimentos a realizar explorando algunos conjuntos de parámetros variables con el fin de analizar su impacto en el sistema. Finalmente se definen las métricas recolectadas para la evaluación de los resultados.

### 5.1. Topologías

Se definieron varias topologías de interés, probando cómo funciona el sistema en las distintas situaciones. En todas las topologías se consideró tanto el tráfico DL como el UL; además, se varió la cantidad de UE conectadas y el tipo de tráfico que pasa por la red. Las topologías son ejemplificadas mediante la interfaz gráfica de OMNeT++, donde tanto la DU y CU son ilustradas como antenas. Esto se debe a que, como fue mencionado en el capítulo anterior, ambas son implementadas por un mismo módulo, sin embargo la conexión con el usuario se da únicamente a través de la DU.

#### **Topología 0**

Con el objetivo de validar el funcionamiento del *Split Manager* se utilizó una topología sencilla en la que se incluye una única radiobase a la que se encuentran conectados cierta cantidad de clientes. Esta topología fue implementada tanto en el simulador base Simu5G como en la biblioteca creada utilizando una opción de *split* en la cual el equipo realiza la totalidad del procesamiento. Se incluye un servidor, que representa una fuente o destino de tráfico de Internet conectado al núcleo de la red (UPF) a través de un enrutador. El tráfico es generado entre el conjunto de usuarios, cuya cantidad puede ser variada, y el servidor.

#### **Topología 1**

Para analizar las diferentes opciones de *splits* implementadas se modificó la topología anterior dividiendo el punto de acceso en un par CU-DU conectados mediante un enlace ideal. De igual manera se tiene un conjunto de usuarios conec-

## Capítulo 5. Experimentación

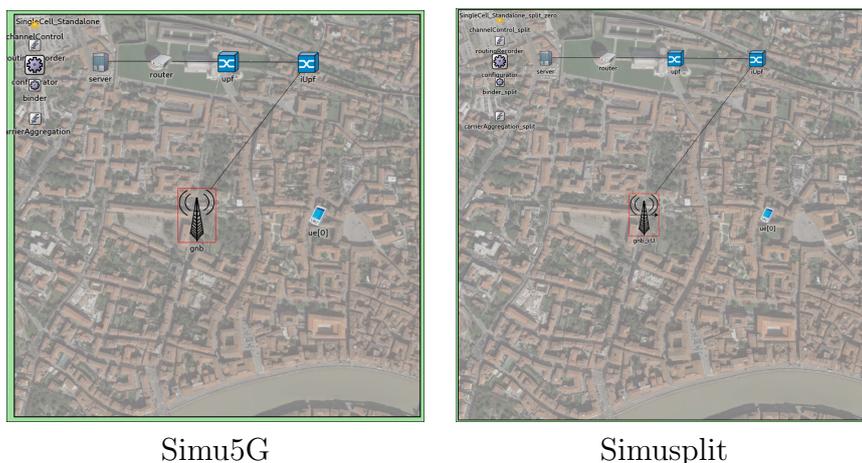


Figura 5.1: Topología 0. En esta topología se cuenta con un conjunto de UE y un servidor conectados mediante una gNB sin utilizar la división DU-CU.

tados a la unidad distribuida los cuales generan o reciben tráfico de un servidor externo.

### Topología 2

Se considera una extensión de la Topología 1, ahora agregando 3 DU conectadas al mismo CU y más UE.

### Topología 3

En esta topología se intenta simular la comunicación de dos UE que se encuentran separados a una distancia considerable. Cada uno de los UE está conectado a su propio par DU-CU. El tráfico debe ser enrutado a través del núcleo de la red (UPF), se considera que este se encuentra localizado en un mismo sitio junto a las CU, como podría ser un *datacenter*. La conexión con las DU hasta las zonas en las que se encuentran los usuarios se realiza a través de un enlace óptico utilizando los módulos del paquete Optical Network —explicado en el capítulo 3.5.

## 5.2. Tipos de Tráfico

Simu5G implementa tres generadores de tráfico básicos que permiten simular distintos tipos de usuario. A continuación se describen los parámetros configurables de cada uno.

- *Constant Bit Rate (CBR)*: Genera tráfico constante a una tasa fija. Los parámetros configurables son:

- Tamaño de los paquetes en bytes.
- Tasa de envío en paquetes por segundo.



## Capítulo 5. Experimentación

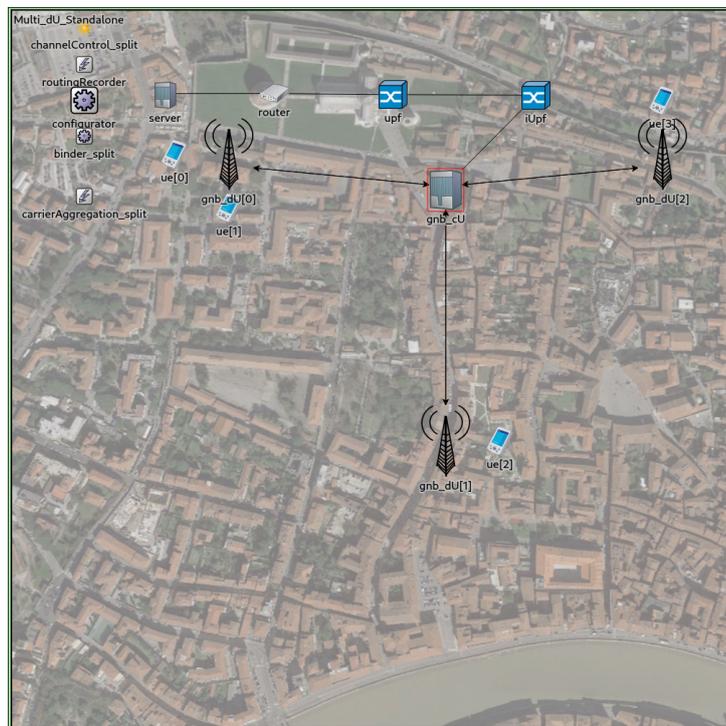


Figura 5.3: Topología 2. En esta topología se cuenta con los UE y un servidor en una comunicación a través mediante una CU y 3 DU sobre las cuales se distribuyen los usuarios.

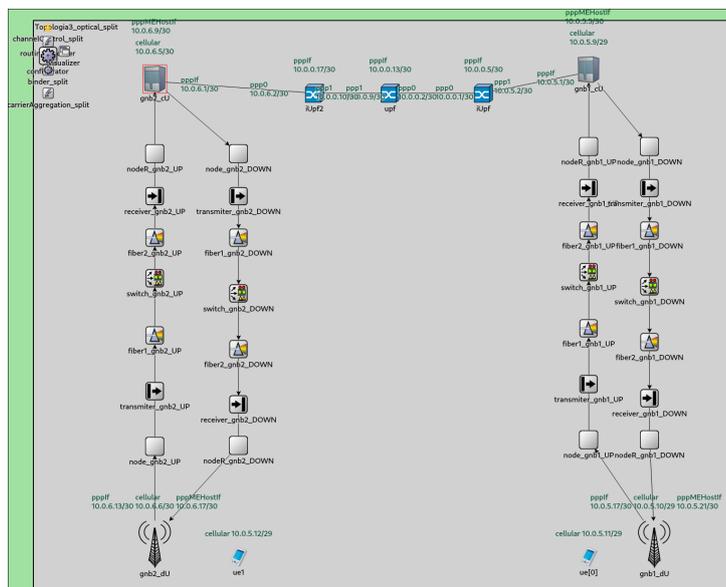


Figura 5.4: Topología 3. Se tienen dos UE comunicándose entre ellos. Cada UE está conectado a un par DU-CU distinto. La conexión entre los DU y las CU se implementa con fibra óptica.

- Parámetro de forma de la distribución Weibull para los intervalos de silencio.
- Parámetro de escala de la distribución Weibull para los intervalos de silencio.

## 5.3. Simulaciones

En primer lugar se define el conjunto de parámetros iniciales, algunos de los cuales se mantuvieron fijos para todas las simulaciones. Luego se describen los experimentos a realizar, para cada uno se realiza un estudio de ciertas combinaciones de parámetros. Para evitar resultados sesgados por la aleatoriedad de las simulaciones, se realizaron 5 repeticiones de cada caso de forma de poder asegurarse de que los resultados obtenidos son consistentes. Para correr las simulaciones de manera automatizada se implementaron *scripts* en Python que permiten la configuración de los parámetros de las simulaciones, su ejecución y la recolección de los resultados. Su uso se detalla en el Apéndice B así como la configuración para la recolección de estadísticas.

### 5.3.1. Parámetros Iniciales

- Cantidad de RB disponibles: 50
- Potencia de transmisión de los UE: 20 dBm
- Potencia de transmisión de los gNB: 40 dBm
- Numerología : 0
- Modelo de Canal: NRChannelModel\_3GPP38\_901 (Siguiendo las especificaciones de [3])
- Planificador: Max C/I
- Ubicación de los UE: Distribución de anillo a una distancia de 300 m del DU
- Tamaño en paquetes de la cola de espera de los equipos CU y DU: 20
- Tiempo de procesamiento de DU:  $50 \mu s$
- Tiempo de procesamiento de CU:  $5 \mu s$
- Tamaño de los paquetes de VoIP: 40 bytes
- Tamaño de los paquetes *burst*: 400 bytes

### 5.3.2. Validación con Simu5G

Para validar que la implementación del *SplitManager* mantuviera el comportamiento original de Simu5G, se realizó un experimento similar a los utilizados más adelante para analizar los diferentes puntos de *split*. Este consistió en variar la cantidad de usuarios conectados a la gNB, tanto para la biblioteca generada como para Simu5G. Aquí se espera comprobar que las métricas evaluadas para ambos simuladores coincidan, implicando que el *SplitManager* no alteró el comportamiento ya validado de Simu5G. La validación se realizó utilizando el tráfico *burst* debido a que, a diferencia de VoIP, ya cuenta con la métrica de *delay* en la implementación original de Simu5G.

### 5.3.3. Variación de cantidad de usuarios

Se varía la cantidad de usuarios conectados a la red, manteniendo los parámetros de cada tipo de tráfico fijos. Se busca analizar cómo se comporta el sistema al aumentar la cantidad de usuarios y en función del tipo de tráfico. Se consideran los tipos de tráfico VoIP y Burst, analizando cada una de las cuatro opciones por separado incluyendo ambos sentidos para cada aplicación. También se analiza un escenario de usuarios mixtos en los cuales se ubica un grupo de usuarios con cada una de las opciones de tráfico y sentido.

### 5.3.4. Tiempo de procesamiento de los equipos

Se varía el tiempo de procesamiento de los equipos DU y CU, lo cual busca analizar como influye la capacidad de cómputo de los equipos en el rendimiento de la red. En [32] se argumenta que la mayoría del tiempo de procesamiento del stack de protocolos radica en las funciones asociadas a la capa física. Dado que para las opciones de *split* implementadas esta se encuentra siempre en el DU, se definió que este sea quien tenga un mayor tiempo de procesamiento. Otro motivo para ello es que la CU típicamente se encuentra en un *datacenter*, donde se dispone de mayor capacidad de cómputo. Además, el mismo trabajo muestra que dichos tiempos de procesamiento son del orden de los microsegundos.

### 5.3.5. Interferencia de otras radiobases

Se simula la presencia de otras radiobases en la zona, las cuales generan interferencia en la comunicación de los usuarios. Con esta se busca provocar una degradación en la calidad de la comunicación, lo que implica la necesidad de retransmisiones. Se espera que aquellas opciones de *split* que centralicen este mecanismo posean un mayor *delay* debido a la necesidad de enviar nuevamente el paquete a través de la red. Los parámetros de las celdas interferentes son:

- Cantidad de celdas: 2
- Potencia de transmisión: 40 dBm

- Cantidad de RB disponibles: 50 por radiobase
- Cantidad de usuarios por celda: 15
- Tipo de tráfico: CBR
- Tamaño de los paquetes: 40 bytes
- Intervalo entre paquetes: 0.01 s

### 5.3.6. Variación del tamaño de paquete

Se investiga el incremento del tamaño de los paquetes, en particular para analizar el comportamiento del sistema al tener que fragmentarlos en varias transmisiones. Las diferentes opciones de *split* presentan una variación de la cantidad de paquetes que deben ser enviados por el *fronthaul* dado que la segmentación se encuentra presente en capas específicas del stack. En particular para el tráfico de tipo *burst* ya que este es un buen modelo para el tráfico de Internet de un usuario típico, donde es de esperar una mayor demanda en términos de capacidad de la red.

- Tamaño de los paquetes de VoIP: 40, 400 y 4000 bytes
- Tamaño de los paquetes de Burst: 400 y 10000 bytes

### 5.3.7. Comparación de la distribución de usuarios en cada DU

Se evalúa la topología 2 consistente de una misma CU conectado a 3 DU diferentes, cada uno con un conjunto de usuarios conectados. Se busca analizar como se comporta la carga en el sistema en función de la distribución de los usuarios en las celdas. Se comparan dos escenarios; uno en el cual los usuarios se dividen equitativamente entre los diferentes equipos DU y otro en el cual se concentran en uno en particular. El escenario consiste de 150 usuarios con las siguientes distribuciones:

- Distribución uniforme: 50 UE asignados a cada equipo DU
- Distribución concentrada: 100 UE asignados al DU número 1 y 25 UE asignados a cada DU restante

### 5.3.8. Comparación de cantidad de usuarios en distintas celdas conectadas mediante red óptica

Se evalúa una comunicación entre dos usuarios conectados a conjuntos de pares CU-DU distintos, conectados mediante una red óptica con el núcleo de la red. Para ello se realiza un estudio similar al realizado en [40], donde se implementa la traducción entre la red móvil y el simulador óptico. Este consiste en un grupo de usuarios conectados a un mismo equipo DU los cuales se encuentran enviando

tráfico de tipo VoIP a un usuario conectado en otro sitio a otro equipo DU. Se evalúa una cantidad de usuarios de 1, 40 y 80.

### 5.4. Estadísticas

Las diferentes simulaciones fueron analizadas en base a la recolección de las siguientes métricas:

- *Delay*: Se tomó el *delay* extremo a extremo a nivel de aplicación, es decir, el tiempo que tarda un paquete en ser enviado desde el UE hasta el servidor y viceversa. Para ello se utilizó la métrica implementada en [40] para el tráfico VoIP, mientras que el tráfico *burst* ya contaba con una medida análoga.
- Tamaño de la cola de paquetes: Esta es medida en los equipos DU y CU, y representa el tamaño en paquetes en la cola de espera antes de ser procesados. Esta muestra la influencia de la sobrecarga de los sistemas en los retardos incluidos en la comunicación.

Si bien el *throughput* de la interfaz CU-DU es una medida de interés, como se observó en el Capítulo 2, para las opciones de *split* implementadas no se espera una gran variación ni un gran requerimiento de este. Por otro lado, su medición requeriría una implementación precisa de los tamaños de los paquetes intercambiados. Simu5G implementa la segmentación de los paquetes en las diferentes capas del stack de protocolos, y asigna de forma correcta su tamaño. Sin embargo algunos paquetes que son enviados para funcionalidades de control, como por ejemplo el pedido de datos de la capa MAC al *buffer* RLC, los cuales no poseen modelado su tamaño y se les asigna un tamaño de 1 byte. Además, para un modelado preciso también se debería implementar un protocolo de comunicación que tenga en cuenta el *overhead* a agregar, así como también la posibilidad de enviar más de un mensaje en una misma transmisión.

Para presentar los resultados se utilizaron las siguientes gráficas:

- Gráficos de violín: Este fue utilizado para presentar los resultados en los cuales se evalúa un único valor, típicamente un promedio, para cada simulación realizada. Esta permite representar la diferentes realizaciones de una misma configuración con diferentes semillas para la generación de números aleatorios. Se indica la mediana del conjunto de datos así como el intervalo correspondiente al percentil 75 para no considerar datos con un comportamiento no típico.
- Gráficos temporales: En algunos casos se presenta el gráfico que muestra el valor registrado de una determinada medida para diferentes instantes de tiempo a lo largo de la totalidad de la simulación. En este caso se grafican todas las repeticiones de una misma simulación con un color distinto y una opacidad baja para poder observar el comportamiento de cada una de ellas, aunque típicamente son muy similares.

## 5.4. Estadísticas

Ya teniendo definidas las distintas topologías, experimentos y parámetros de las simulaciones, en el siguiente capítulo se analizan los distintos resultados obtenidos en la herramienta.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 6

## Análisis de resultados

En este capítulo se analizan las estadísticas recolectadas en función de cada opción de *split* implementada en la herramienta. Si bien se realizó una gran cantidad de experimentos, se presentan aquí únicamente aquellos que son considerados relevantes. Algunas de los no incluidos se pueden consultar en el Apéndice D. Por otra parte en el Apéndice C se encuentran los tiempos requeridos para ejecutar las simulaciones así como información del ambiente en el que se corrieron.

### 6.1. Topología 0

En primer lugar, se validó que el simulador con la inclusión del *SplitManager* mantuviera el comportamiento original de Simu5G al utilizar una configuración donde la totalidad del procesamiento se realiza en el DU, lo que denominamos *split 0*. El funcionamiento original no modela de la misma manera la capacidad de procesamiento de los nodos (tiempo de procesamiento, *buffers* finitos). Debido a esto se realizó la validación con una configuración sin retardo asociado al tiempo de procesamiento, así como una cola de paquetes de tamaño infinito de forma de no descartar paquetes. La métrica evaluada para la validación es el *delay* extremo a extremo. En la figura 6.1 se observa la gráfica de violín promediada sobre el conjunto de usuarios y variando su cantidad. Los experimentos mantienen los mismos resultados en ambas bibliotecas, lo que implica que la inclusión del *SplitManager* no influye en el comportamiento asociado a esta métrica.

### 6.2. Topología 1

Los experimentos en esta sección fueron realizados sobre la topología de la Figura 6.2 donde se tiene una DU y una CU, conectadas con un enlace ideal. Esta conexión no introduce retardos ni errores.

## Capítulo 6. Análisis de resultados

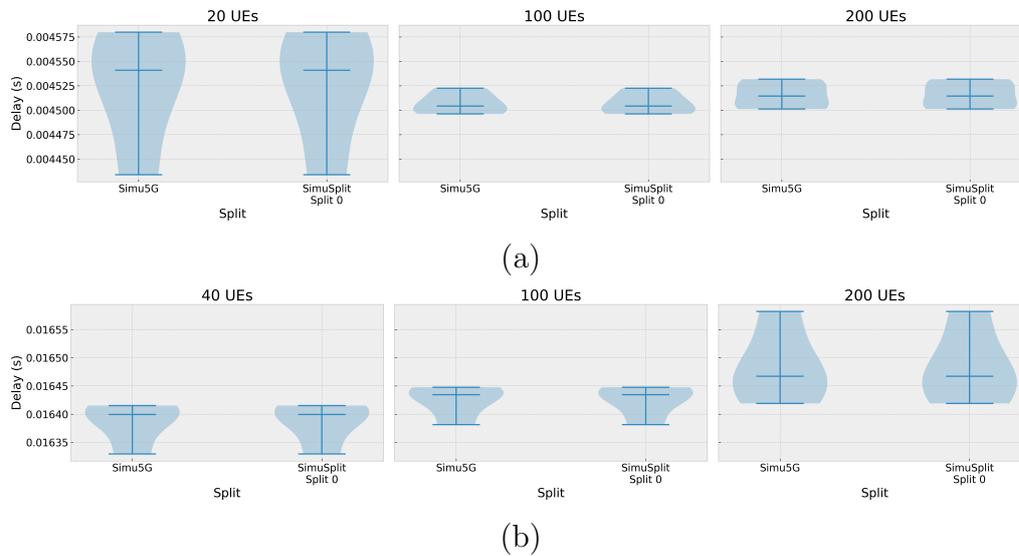


Figura 6.1: Gráfica de violín para las métricas de *delay* extremo a extremo con usuarios de tráfico burst en sentido DL (a) y UL (b). Cada columna corresponde a una simulación con una determinada cantidad de usuarios y se compara la simulación realizada en Simu5G y la biblioteca implementada.

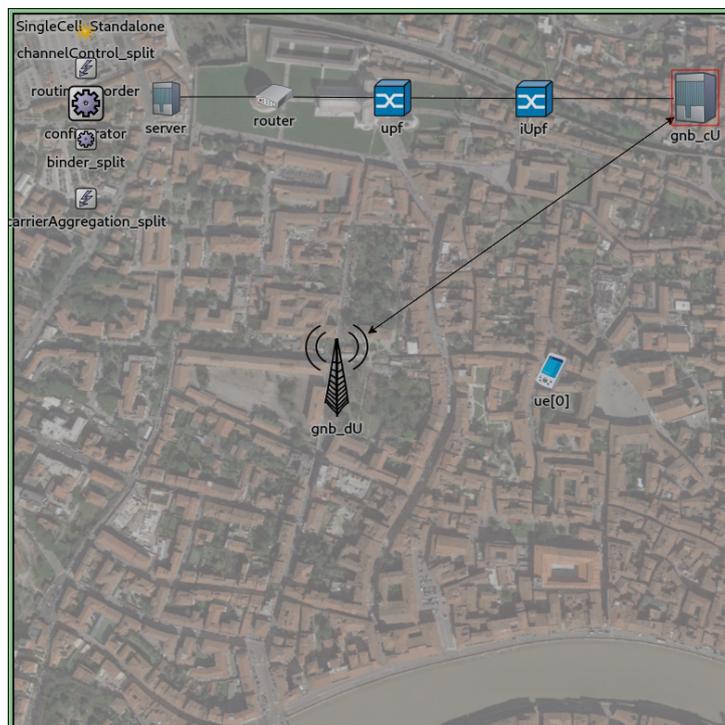


Figura 6.2: Topología 1. En esta topología se cuenta con un conjunto de UE y un servidor conectados mediante una CU y una DU. El enlace entre las dos unidades es ideal; no introduce errores ni retardo.

### 6.2.1. Comparación de cantidad de usuarios

Este primer experimento analiza cómo se comporta el sistema al variar la cantidad de usuarios en la simulación, lo cual impacta en la congestión y saturación de los recursos disponibles. Esta cantidad finita de recursos se representa en la simulación con el tamaño de la cola de paquetes presentes en los equipos, lo que se traduce en un tiempo de espera hasta ser atendidos. Se espera que al aumentar la cantidad de usuarios los *splits* más altos que centralizan más funciones presenten una mayor latencia debido a resolver localmente una menor cantidad de mensajes.

Se evaluó el escenario con la totalidad de los usuarios en la simulación pertenecientes a un único grupo de tráfico. En la Figura 6.3 se muestra el gráfico de violín del *delay* extremo a extremo para el tráfico VoIP en sentido DL así como un gráfico de violín de los paquetes perdidos para cada caso, ambas promediadas sobre la totalidad de los usuarios para cada repetición. Según [17] para una experiencia satisfactoria en una llamada VoIP se requiere una latencia menor a 100 ms, y entre 100 a 400 ms es viable pero notorio. Se observa que a partir de 40 usuarios, la cantidad de paquetes perdidos en la opción de *split 6* comienza a incrementar, al punto de no ser tolerable para 100 o más usuarios. Por otro lado la única latencia que se ve incrementada es la correspondiente al *split 4* con 200 usuarios, esto refleja el requisito estricto de latencia necesario para los *splits* altos. Si bien los paquetes que logran ser recibidos para el *split 6* presentan una baja latencia, una gran cantidad son perdidos ya que la capa MAC en el equipo centralizado no logra realizar sus funcionalidades en los tiempos marcados por la estructura de la transmisión. Esto lo convierte en una opción no viable para casos de uso que requieran una alta confiabilidad en la transmisión de paquetes. En la Figura 6.4 se observa la misma gráfica para el caso en el que los usuarios realizan tráfico en UL. Se tiene un comportamiento del *split 6* similar, en el que para una cantidad de usuarios de 100 o 200 el sistema pierde más paquetes de lo tolerado. Sin embargo para el caso del *split 4* en este caso no se da un incremento del *delay* extremo a extremo, con lo cual el sentido UL se ve menos afectado por la congestión. Esto se debe a que en este caso la capa RLC únicamente almacena los paquetes recibidos desde la capa MAC en su *buffer*. En sentido DL la capa RLC debe notificar que hay tráfico disponible, se debe realizar la asignación de recursos y posteriormente solicitar dichos datos para enviar, esta mayor cantidad de pasos incrementa la latencia.

En la Figura 6.5 se muestra el tamaño promedio de la cola de paquetes en el equipo CU y DU para una simulación con 100 UE y comparando un escenario con tráfico VoIP y *burst* en sentido UL. El tráfico *burst* con *split 6* congestiona más esta cola de espera en el equipo CU en comparación al tráfico VoIP, lo cual es razonable ya que este necesita traficar volúmenes mayores en intervalos pequeños de tiempo lo cual sobrecarga más los recursos. En particular esta opción de *split* requiere traficar a través del *fronthaul* los paquetes de la capa MAC donde se necesita realizar la asignación de recursos, esto implica una mayor carga en el equipo. Se aprecia la carga en el equipo DU crece para el tráfico VoIP pero no en el caso de *burst*, esto puede deberse a que el mecanismo HARQ debe ser realizado para cada paquete más pequeño de manera independiente. El tráfico en rafagas

## Capítulo 6. Análisis de resultados

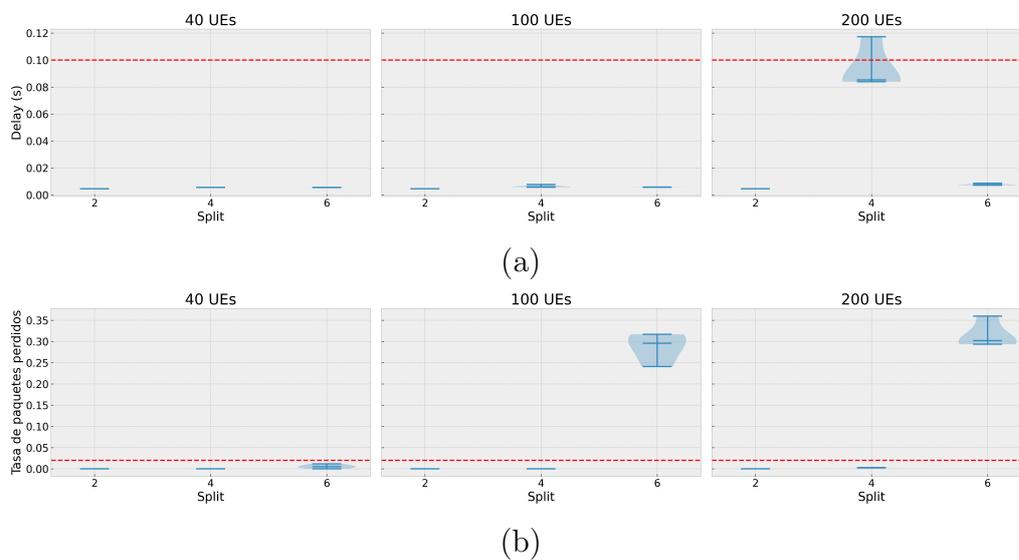


Figura 6.3: Gráfica de violín para las métricas de *delay* extremo a extremo (a) y tasa de paquetes perdidos en una simulación con usuarios de tráfico VoIP en sentido DL (b) promediadas sobre todos los usuarios. Cada columna corresponde a una simulación con una determinada cantidad de usuarios y se comparan las tres opciones de *split* implementadas.

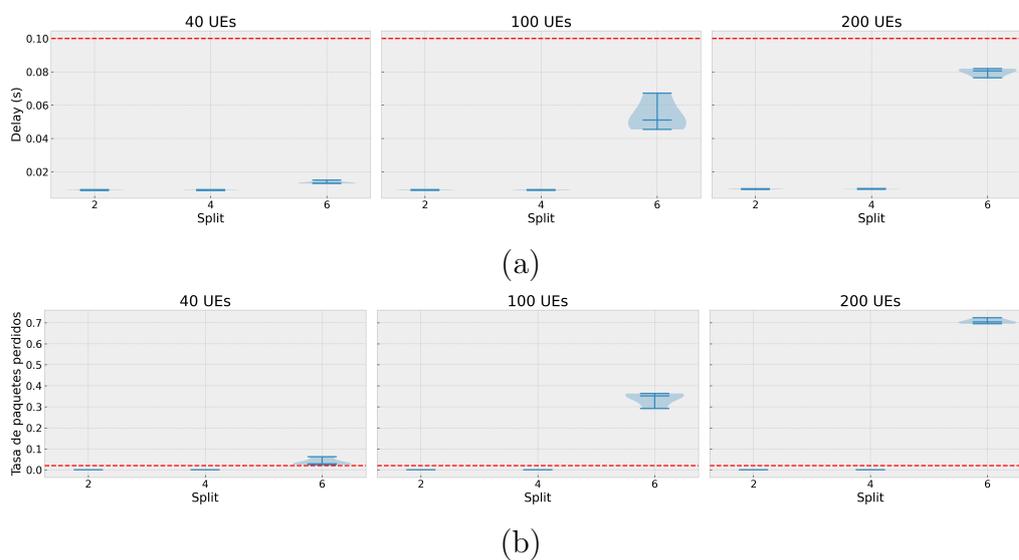


Figura 6.4: Gráfica de violín para las métricas de latencia extremo a extremo y paquetes perdidos en una simulación con usuarios de tráfico VoIP en sentido UL, ambas magnitudes se promedian sobre todos los usuarios. Cada columna corresponde a una simulación con una determinada cantidad de usuarios y se comparan las tres opciones de *split* implementadas.

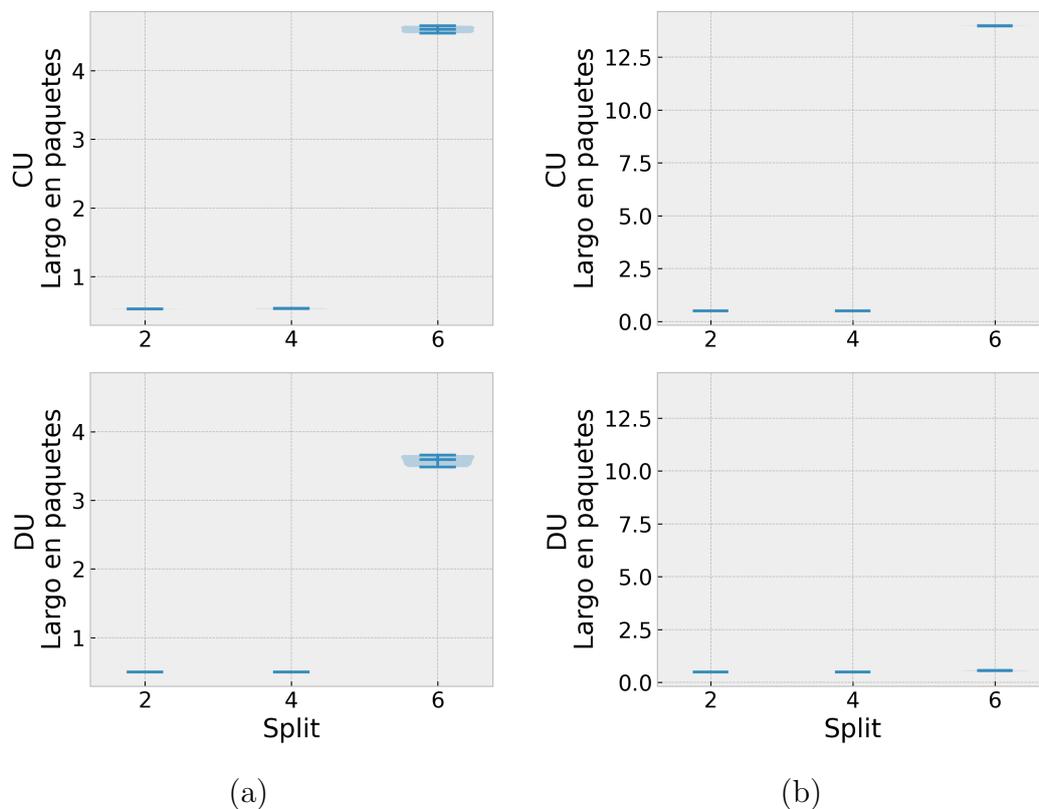


Figura 6.5: Gráfica de violín para el tamaño promedio de la cola de paquetes en el equipo CU y DU en una simulación con 100 usuarios de tráfico VoIP en sentido UL para tipos de tráfico VoIP (a) y *burst* (b). Se comparan las tres opciones de *split* implementadas.

puede ser confirmado de una única vez, ya que al producirse en intervalos pequeños es posible que puedan ser enviados en un mismo intervalo de transmisión.

En la Figura 6.6 se muestra el *delay* promediado sobre todos los clientes de extremo a extremo para distintas cantidades de usuarios en la simulación, distinguiendo la latencia vista por cada grupo de tráfico.

Dado que típicamente el equipo centralizado se encuentra en un entorno virtualizado, es posible escalar de forma dinámica la cantidad de recursos disponibles. Por tanto, aquellos escenarios en los que se observa una sobrecarga de la CU en un escenario real podrían ser mitigados con una detección apropiada que permita incrementar la capacidad de procesamiento.

### 6.2.2. Interferencia de otras radiobases

Al introducir interferencia de otras radiobases en la simulación se incrementa el número de retransmisiones necesarias para completar la transmisión de un paquete. En la Figura 6.7 se muestra nuevamente el gráfico comparativo de los *delays* observados en distintas repeticiones de la simulación. Se diferencia entre tipos de usuarios y se varía su cantidad manteniendo una distribución equitativa

## Capítulo 6. Análisis de resultados



Figura 6.6: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios mixtos. Cada columna corresponde a una simulación con una determinada cantidad de usuarios por grupo y se comparan las tres opciones de *split* implementadas. Se muestra el *delay* separado por tipo de usuario.

entre ellos.

En primer lugar y como es de esperar, la latencia resultante es mayor que en el experimento previo en el cual no se daban una cantidad significativa de retransmisiones, ya que algunos paquetes deben recorrer ciertos enlaces más de una vez. Estos resultados también presentan una mayor variación en diferentes repeticiones ya que los resultados dependen de más factores, en este caso el tráfico de los usuarios en diferentes celdas. Por otro lado una diferencia significativa respecto al escenario anterior es que la latencia percibida por los usuarios en sentido DL no es despreciable, y se encuentra con valores similares o mayores a la latencia en sentido UL. Esto se debe a que a que el modelo realizado para los recursos de procesamiento se encuentra centrado en la DU y CU, por lo que son los paquetes

## 6.2. Topología 1

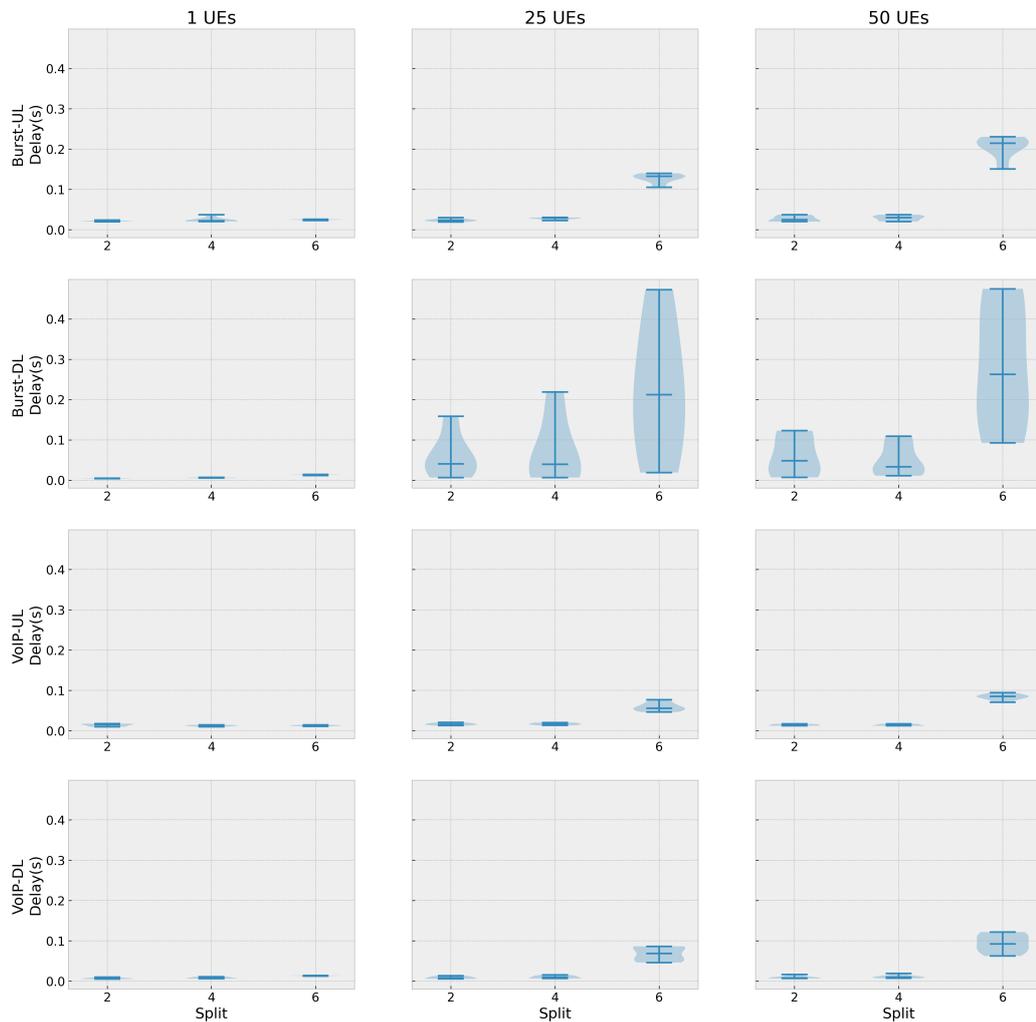


Figura 6.7: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de mixtos e interferencia de otras radiobases. Cada columna corresponde a una simulación con una determinada cantidad de usuarios por grupo y se comparan las tres opciones de *split* implementadas. Se muestra el *delay* separado por tipo de usuario.

retransmitidos por estas los que se ven más afectados por este modelo. Finalmente, respecto al punto de *split* se observa que el *split* 6 presenta un *delay* considerablemente más alto al resto. Esto es explicado por la presencia de la capa MAC en el equipo centralizado, esta etapa contiene el mecanismo de retransmisiones HARQ y sobrecarga el sistema al no realizarse localmente debido a la necesidad de las retransmisiones de atravesar el *fronthaul* nuevamente.

Si bien en este las retransmisiones son debidas a la interferencia de otras estaciones, es de esperar que se de un comportamiento similar en un escenario real en el que la interferencia sea debida a otros factores como la atenuación de la señal o interferencia proveniente de otros dispositivos.

## Capítulo 6. Análisis de resultados

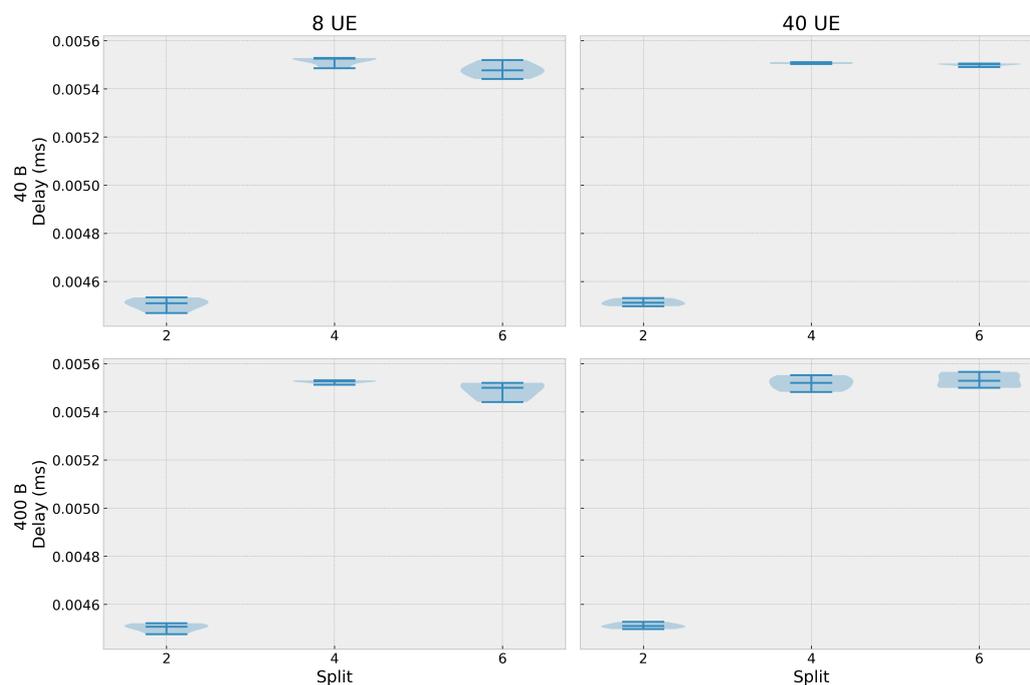


Figura 6.8: Gráfica de violín para las métricas de latencia extremo a extremo en una simulación con VoIP en sentido DL. Cada columna corresponde a una simulación con una determinada cantidad de usuarios mientras que cada fila corresponde a un tamaño de paquete enviado, se comparan las tres opciones de *split* implementadas.

### 6.2.3. Variación del tamaño del paquete

Para el caso en el que los usuarios son todos del tipo VoIP en sentido UL se observó que si se utiliza un paquete de tamaño 4000 bytes el sistema se ve saturado, presentando retardos y cantidad de paquetes perdidos altos.

En las Figuras 6.8 se observa la comparación de la latencia recibida promedio por los usuarios para dos cantidades de UE, 8 y 40, con dos tamaños de paquetes 40 y 400 en sentido DL. La misma información se muestra en 6.9 para el sentido UL. Si bien no se observan grandes diferencias en función del tamaño utilizado, se puede notar que en sentido UL se presenta un incremento de la latencia al aumentar el tamaño del paquete mientras que para el sentido DL se mantienen los valores. En el Apéndice D se incluyen las gráficas asociadas al tráfico *burst* donde se observa un comportamiento similar.

Un posible caso de uso en el que se transmiten paquetes de tamaño mayor con un requerimiento estricto de latencia son los videojuegos en la nube o comunicaciones asociadas a la realidad virtual o aumentada. Estas aplicaciones requieren de una latencia de extremo a extremo de 5 a 50 ms para ser consideradas aceptables según [5]. El tipo de tráfico generado por estas aplicaciones puede ser modelado como tráfico de tipo *burst*. Por otro lado aplicaciones de videoconferencia o *streaming* de video en vivo pueden ser modeladas utilizando el perfil de VoIP con tamaños de paquete mayores, principalmente debido a la transmisión de cuadros de video.

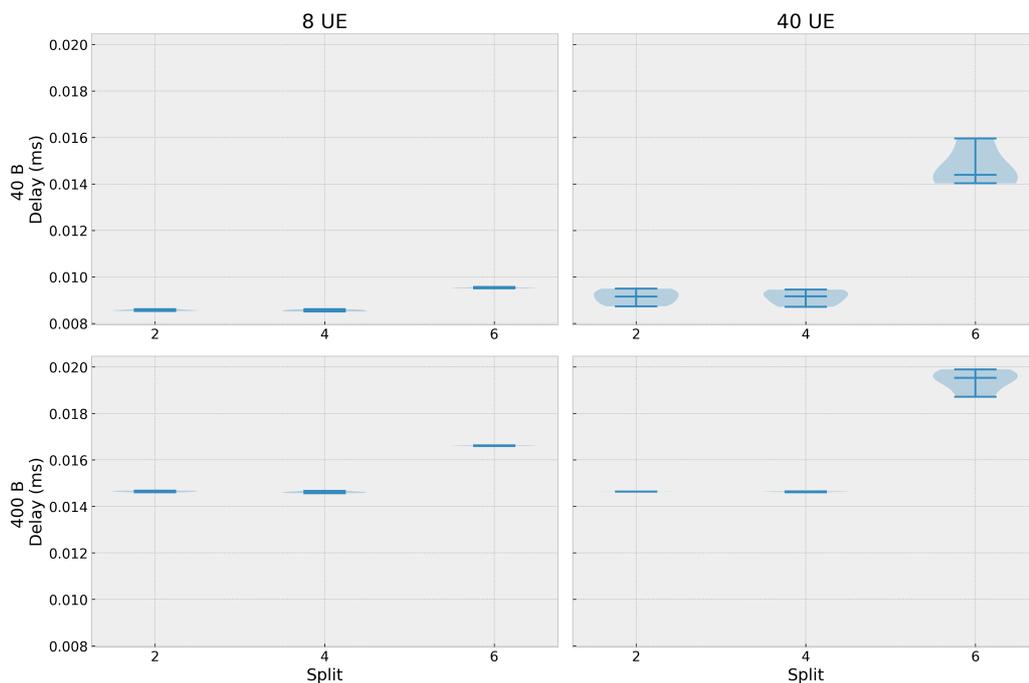


Figura 6.9: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios mixtos. Cada columna corresponde a una simulación con una determinada cantidad de usuarios por grupo y se comparan las tres opciones de *split* implementadas. Se muestra la latencia separado por tipo de usuario.

Para sistemas de videoconferencia la latencia, similar al caso de una llamada, debe encontrarse en el entorno de los 150 ms.

## 6.3. Topología 2

La topología correspondiente a este experimento, ilustrada en la Figura 6.10, contiene tres DU conectadas a un mismo CU. Además cuenta con un servidor que se comunica con los usuarios de forma análoga a los escenarios anteriores.

### 6.3.1. Comparación de distribución de usuarios por celda

En la Figura 6.11 se puede ver una comparación del promedio de paquetes total en las colas de espera del equipo CU al atender a tres equipos DU con usuarios de tipo *burst* en sentido UL. Se compara una distribución uniforme sobre los tres equipos DU así como una distribución en la que los usuarios se concentran en uno en particular. Se puede apreciar que la distribución uniforme genera una mayor cantidad de paquetes en promedio en espera para las opciones de *split* 2 y 4 en comparación a la distribución concentrada. Esto parece razonable ya que concentrar los usuarios en un mismo equipo DU permite por un lado aprovechar las comunicaciones con el equipo CU que son comunes a todos los usuarios y por

## Capítulo 6. Análisis de resultados

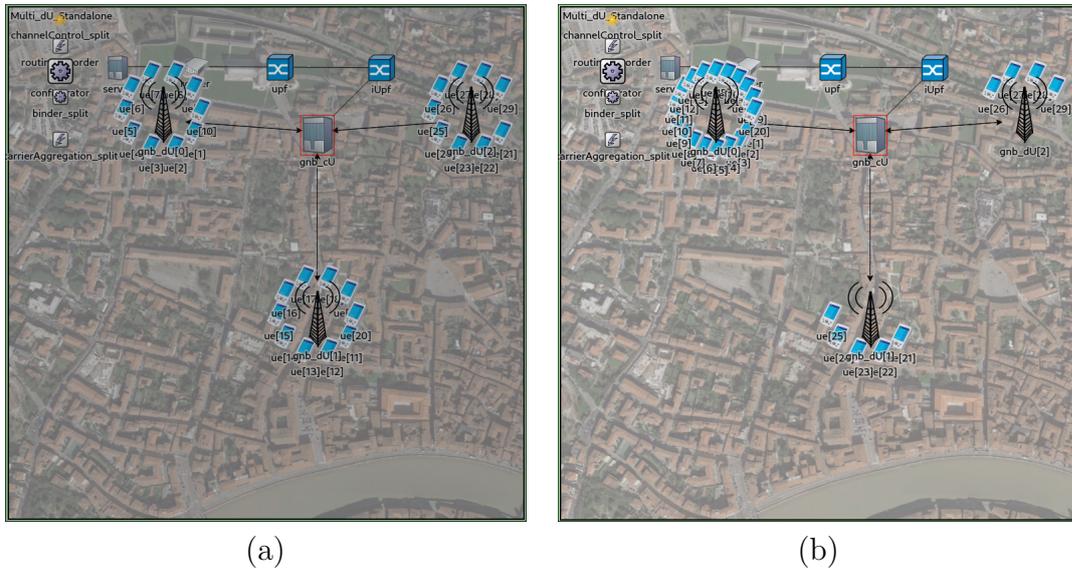


Figura 6.10: Topología 2. En esta topología se tienen 3 DU conectados a una misma CU. Se muestran los dos escenarios comparados, la totalidad de los UE distribuidos uniformemente entre las tres DU (a) y concentrados en una particular (b).

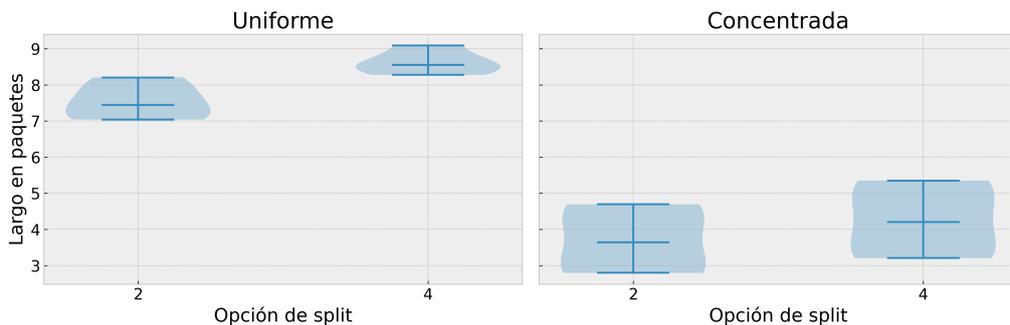


Figura 6.11: Gráfica de violín para la cantidad de paquetes promedio en las colas de espera del equipo CU conectado a tres DU en una simulación con tráfico *burst* en sentido UL. Se comparan dos distribuciones de cantidad de usuarios sobre los equipos DU, y se analizan las dos opciones de *split* implementadas para este escenario.

tanto pueden ser manejadas en conjunto, como la asignación de recursos. Además el concentrar los usuarios en un mismo DU puede sobrecargarla, con lo cual su tráfico se ve limitado pero sin afectar el de otros nodos. En la Figura 6.12 se observa la misma comparación para el caso de tráfico en DL. En este caso también se da un incremento del tamaño de la cola de paquetes del CU para el escenario con los usuarios uniformemente distribuidos entre las DU, sin embargo esto es únicamente para el *split* 4. Esto se debe a que la capa RLC, presente en la CU en dicha opción, debe almacenar en su *buffer* los paquetes para transmitir hasta que estos tengan recursos asignados para ser enviados.

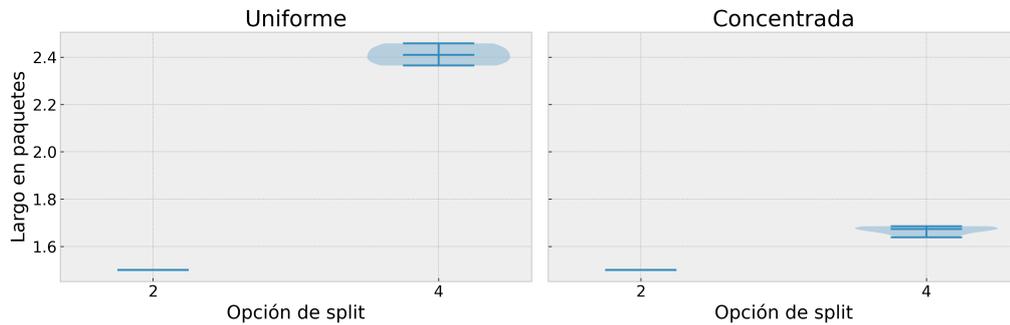


Figura 6.12: Gráfica de violín para la cantidad de paquetes promedio en las colas de espera del equipo CU conectado a tres DU en una simulación con tráfico *burst* en sentido DL. Se comparan dos distribuciones de cantidad de usuarios sobre los equipos DU, y se analizan las dos opciones de *split* implementadas para este escenario.

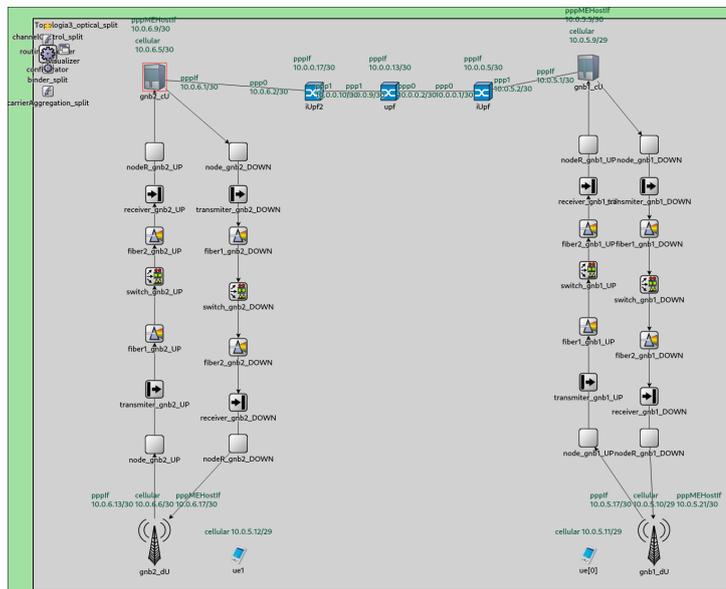


Figura 6.13: Topología 3. Se tienen dos UE comunicándose entre ellos. Cada UE está conectado a un par DU-CU distinto. La conexión entre las DU y las CU se implementa con fibra óptica.

## 6.4. Topología 3

Se trabajará en esta sección con la topología ilustrada en la Figura 6.13. En esta se hallan dos UE conectados a distintos pares DU-CU. El enlace entre cada DU y su CU está hecho con fibra óptica de la biblioteca Optical Network (explicado en la sección 3.5)

### 6.4.1. Comparación de la cantidad de usuarios

Se muestra en la Figura 6.14 la comparación de distinta cantidad de usuarios transmisores enviando tráfico VoIP a un usuario destino conectado en otra radio-

## Capítulo 6. Análisis de resultados

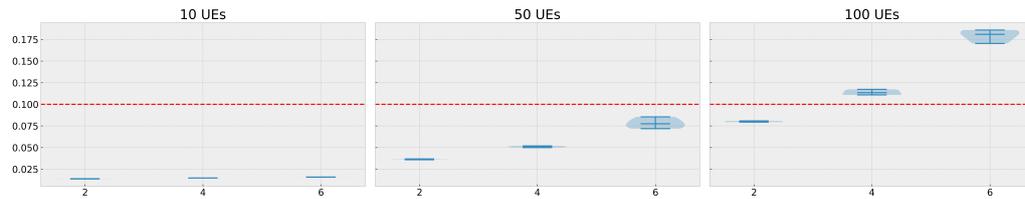


Figura 6.14: Gráfica de violín para la latencia percibida por el usuario receptor en promedio en una simulación con tráfico VoIP. Se comparan distintas cantidades de usuarios transmisores, y se analizan las 3 opciones de *split* implementadas.

base. Se observa que en comparación a los experimentos en los que no se tiene una red óptica que interconecta los sitios, este escenario presenta un *delay* general más alto para todos los *splits*. Este depende de la distancia a recorrer, y por tanto impone restricciones en la distancia máxima para los *splits* altos con requerimientos estrictos de latencia.

Habiendo analizado y evaluado nuestra herramienta, se procede a extraer conclusiones y proponer nuevas líneas de investigación en donde continuar desarrollando y explotando este simulador.

# Capítulo 7

## Conclusiones y trabajo futuro

### 7.1. Herramienta obtenida

El trabajo realizado logró el objetivo planteado de desarrollar una plataforma de simulación que permitiera la evaluación de diferentes opciones de *split* en redes 5G. Esta fue desarrollada sobre una plataforma de código abierto con una gran contribución de la comunidad en paquetes que implementan modelos específicos, como es el caso de Simu5G.

En primer lugar, si bien la biblioteca obtenida cuenta con varios módulos extraídos de Simu5G, se aseguró que no hayan conflictos. Por esta razón, se pueden tener dentro de un mismo proyecto de OMNeT++ o en una misma topología de simulación, módulos de las dos bibliotecas. A su vez, al ser independiente de Simu5G se permite que esta biblioteca, que está en continuo desarrollo, realice actualizaciones sin afectar al funcionamiento de los módulos creados en este proyecto.

A su vez, queda publicado de forma abierta al público el repositorio Git con el trabajo realizado [6]. Dentro de este repositorio se encuentran todos los módulos implementados y modificados, las tres topologías utilizadas para evaluar el desempeño de los puntos de *split*, y los *scripts* que fueron utilizados para la obtención de las gráficas de análisis, los cuales se describen en el Apéndice B. Al ser un repositorio *open source*, queda a disposición de la comunidad científica para su posterior uso y modificación.

Respecto a los objetivos planteados, se consiguió con éxito implementar los *splits* 2, 4 y 6. Todos estos funcionan tanto para tráfico UL como para DL, y a su vez es posible diseñar topologías con más de un DU conectado al mismo CU. Estas funcionalidades permiten que se diseñen una amplia gama de escenarios de interés, y aún más considerando el simulador de redes ópticas Optical Network disponible sobre la misma plataforma. A su vez, al momento de diseñar el módulo *Split Manager*, se consideraron las posibles extensiones naturales de este. Al centralizar el comportamiento asociado a los *splits*, permite que se extienda su funcionalidad al implementar más puntos de *split*, y también simplifica la implementación de lógica de control. Junto con este último punto, ya se crearon protocolos de control para los módulos *Split Manager* que permiten que se modifiquen durante la simulación

## Capítulo 7. Conclusiones y trabajo futuro

algunos parámetros como el punto de *split*.

Por último se validó el simulador obtenido sobre varios escenarios con distintas características y parámetros para evaluar cómo afectan estos en el sistema. Se analizó una métrica de suma importancia para varias de las aplicaciones novedosas que pretende 5G como es la latencia extremo a extremo. También se investigó la tasa de paquetes perdidos en función de la opción de *split* elegida, observando que para las opciones que centralizan más funciones se tiene un requisito más restrictivo sobre el retardo que puede introducir el *fronthaul*. Esto último coincide con lo esperado en base a la bibliografía disponible como se mencionó en el Capítulo 2.

### 7.2. Trabajo a futuro

La herramienta implementada provee una base sólida para la simulación de redes 5G. Sin embargo, existen varias áreas en las que se podría trabajar para mejorar la herramienta y extender su funcionalidad. Esto en parte gracias al diseño de OMNeT++ que facilita su extensión gracias a su diseño modular. A continuación se presentan algunas ideas para trabajo a futuro.

#### 7.2.1. Extender las opciones de *split* soportadas

En primer lugar aún se encuentran pendientes las implementaciones algunas de las opciones de *splits* definidas en por la 3GPP. En particular como ya se mencionó anteriormente, los operadores presentan un gran interés por los *splits* de capa física (7.1, 7.2, 7.3 y 8) que permiten el despliegue de celdas pequeñas y baratas. Esto permite afrontar la gran densidad de puntos de acceso necesarios para frecuencias altas, orientadas a la zonas con alta densidad de usuarios. Para la implementación de dichos *splits* se deberá realizar un modelado de las funciones de capa física de la gNB, o al menos de sus impactos sobre las métricas de interés como son el retardo y el ancho de banda requerido para el *fronthaul*. Este último cobra particular interés en el caso de los *splits* de capa física, ya que algunos de ellos implican la transmisión de muestras temporales en lugar de bits de información útiles lo cual implica un aumento considerable en el ancho de banda requerido.

#### 7.2.2. Asignación dinámica del punto del *split*

Otra área de interés es la implementación de algoritmos de asignación dinámica del punto de *split* con el objetivo de adaptarse a los requisitos y capacidades de la red en tiempo real. El diseño del *Split Manager* permite la configuración de la división de funcionalidades a utilizar por parte del CU. Esto habilita la implementación de un algoritmo de decisión en dicho nodo que envíe los mensajes de configuración necesarios a los DU en base a la información que posea sobre la red. Un futuro trabajo podría dejar definida una interfaz para la implementación de dichos algoritmos así como la evaluación de alguno de los algoritmos propuestos en la literatura. Esto es de interés para los operadores de red ya que es de esperar

que las arquitectura desplegadas tengan la posibilidad de realizar todas las funcionalidades en el sitio de conexión con los usuarios así como en un sitio centralizado. Los recursos cerca de los clientes se utilizan para dar servicios de baja latencia y alta confiabilidad, implementado en hardware específico lo que los hace más caros y exclusivos. Por otro lado, los recursos centralizados se utilizan para dar servicios que no requieren de esta baja latencia, y se pueden implementar en hardware genérico. Ambos servicios pueden ser vendidos a precios distintos, lo que permite a los operadores maximizar sus ingresos. Los algoritmos de asignación dinámica son cruciales para lograr una asignación eficiente de los recursos y en tiempo real.

### 7.2.3. Protocolos en el enlace DU-CU

También se podría considerar la inclusión de algún protocolo de comunicación entre los DU y los CU. En la actualidad, la comunicación entre estos nodos se realiza mediante un enlace directo en el cual los mensajes asociados a cada capa son enviados de forma transparente entre los nodos. Sin embargo podría implementarse la utilización de algún protocolo de comunicación que modele el *overhead* necesario para la comunicación. Relacionado a este punto, podría ser de interés un modelado más certero del tamaño de los mensajes de control y de datos que se envían entre los nodos. En la actualidad los mensajes que no acarrean datos útiles tienen un tamaño constante de 1 Byte. Si bien es razonable considerar que estos tienen un tamaño pequeño en comparación con los mensajes de datos, un análisis más detallado de los tamaños de los mensajes de control y de datos en una red real podría permitir una mejor modelación de la red.

### 7.2.4. Ajuste de parámetros

Lograr un modelo de una red realista con los parámetros disponibles en el simulador, tanto en los provistos por Simu5G como en los introducidos en este trabajo, puede contribuir a validar el sistema de forma más precisa. Especialmente si se tiene acceso a un entorno real, como puede ser una maqueta, para contrastar los resultados obtenidos en simulación y en un experimento práctico.

### 7.2.5. Separación de punto de *split* según cliente

Actualmente, al inicio de la simulación es el CU que determina el punto de *split* de todos los DU a los cuales está conectado. Una posible extensión a la biblioteca creada es hacer que dependiendo del cliente, se utilicen *splits* distintos. A su vez, podría ser de interés, que un mismo CU tenga conectado varios DU con distinto punto de *split*.

### 7.2.6. Impacto de red óptica en puntos de *split*

La extensión del simulador óptico, incluyendo opciones como el multiplexado flexible del espectro, y su impacto en los distintos puntos de *split* es un posible

## Capítulo 7. Conclusiones y trabajo futuro

trabajo futuro. Dado que la conexión entre los DU y CU puede estar hecha con fibra óptica, resulta pertinente realizar un estudio de cómo afecta a la elección del punto de *split* la tecnología utilizada en la fibra óptica y el tipo de conexión entre los DU y CU. A su vez, el dinamismo de algunas técnicas como flexgrid complementa el estudio de la asignación dinámica del punto de *split*.

### 7.2.7. Simulación de canal satelital

Con el resurgimiento de los satélites para las conexiones móviles, puede ser útil para su estudio la implementación de su canal de transmisión. Con esto se podría evaluar el mejor punto de *split* al momento de poner los dispositivos DU y CU en órbita.

# Apéndice A

## Cambios en Simu5G

### A.1. Nuevas puertas

Para varios módulos utilizados en la NIC se crearon puertas de entrada y salida para conectarse con el nuevo módulo Split Manager. Durante la ejecución, se verifica que el módulo corresponda al de una DU o CU, y solo en ese caso se utilizan las nuevas puertas: mandando todos los mensajes por la de salida, y aceptando los mensajes entrantes por la puerta de entrada. De esta manera, el comportamiento de los UE queda incambiado, mientras que es posible conectar el Split Manager a los módulos de la NIC de la gNB.

**IP2NIC:** La nueva puerta de salida es llamada IP2NIC\_to\_SM y la de entrada SM\_to\_IP2NIC.

**PDCP-RRC:** La nueva puerta de salida es llamada PDCP\_to\_SM y la de entrada SM\_to\_PDCP.

**RLC:** La nueva puerta de salida es llamada RLC\_to\_SM y la de entrada SM\_to\_RLC.

**MAC:** La nueva puerta de salida es llamada MAC\_to\_SM y la de entrada SM\_to\_MAC.

**PHY:** La nueva puerta de salida es llamada PHY\_to\_SM y la de entrada SM\_to\_PHY.

#### A.1.1. Comportamiento RLC

En la implementación original de Simu5G el módulo RLC contaba con 3 módulos conectados al PDCP: UM, AM y TM, cada uno utilizando una conexión distinta, y un multiplexador que los conecta a la MAC. A modo de ejemplo un flujo DL podría ser que pase por los bloques: PDCP, UM, MUX, MAC; mientras que

## Apéndice A. Cambios en Simu5G

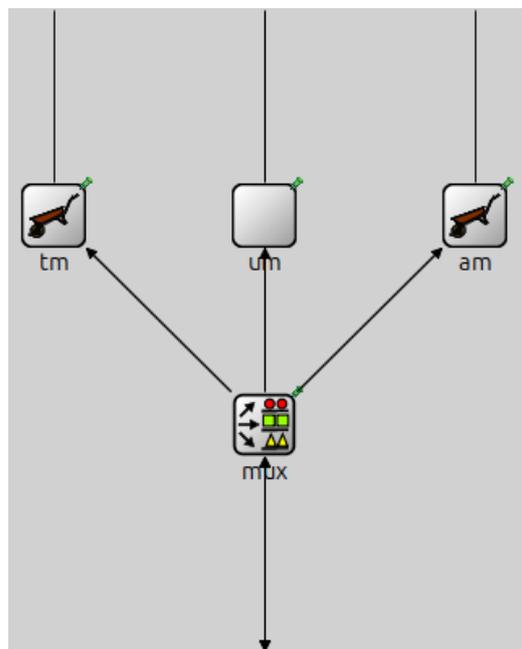


Figura A.1: Implementación de Simu5G del módulo RLC. En dirección UL los submódulos UM, AM y TM, están directamente conectados al PDCP. En dirección DL, el submódulo MUX está conectado a la MAC

un flujo UL podría ser: MAC, MUX, AM, PDCP.

En el módulo Split Manager se optó por tener solamente una conexión de entrada y una de salida a cada módulo, por lo que fue necesario modificar levemente el comportamiento del módulo RLC. En la nueva implementación, todos los mensajes entrantes son recibidos por el MUX y es este el que redirige el paquete a las interfaces correspondientes de los módulos UM, AM y TM. De igual manera, todos los mensajes salientes son recibidos por el MUX y este los envía al Split Manager.

### A.2. Dirección de los Mensajes

Por cómo funciona el Split Manager, es necesario que todos los mensajes cuyo destino sea un módulo más cercano a la capa 3 que el origen, tengan marcada la dirección como UL (este es un parámetro de los tags `UserControlInfo` y `FlowControlInfo`). Si el destino está más cerca a la capa 1, entonces este tendrá que estar marcado como DL. En Simu5G, la configuración por defecto de algunos mensajes no es compatible con los requerimientos del Split Manager. Es por eso que se necesitó alterar la dirección de algunos tipos de mensajes al momento de ser creados, y luego corregir este cambio cuando son leídos.

**HARQ\_Rx:** Se configura para que la nueva dirección sea la opuesta a la del flujo de mensajes.

### A.3. GtpUser / TxPDCP (UE DestId)

**MacSduRequest:** Durante el proceso de solicitud de recursos por parte de la MAC al RLC, no todos los mensajes tenían la dirección correcta.

### A.3. GtpUser / TxPDCP (UE DestId)

El comportamiento nativo de Simu5G crea el túnel GTP-U con el destino del paquete como la gNB a la cual se encuentra conectado el UE. Sin embargo, dado que ahora funciones asociadas al túnel se encuentran en el CU el cual difiere del DU al que se encuentra conectado el UE por la interfaz de radio, es necesario modificar la terminación del túnel GTP-U en el CU. Este cambio fue realizado en el modulo GtpUser, donde se modificó el destino en caso de que la radiobase a la cual se encuentra conectado el UE se trate de un DU.

### A.4. Comportamiento IP2NIC para los UE no conectados al CU

Similar al punto anterior, la capa IP2NIC de la gNB descarta los paquetes destinados a los UE que no se encuentran conectados a ella. Dado que el CU debe recibir los paquetes de los UE conectados a los DU que controla, se deshabilitó dicho control de forma que los paquetes no sean descartados.

### A.5. MAC

Se modificó el comportamiento de la capa MAC para que este envíe los paquetes a la capa PHY en caso de tenerse un delay entre la solicitud de los paquetes a la capa RLC, como es el caso del *split* opción 4.

Esta página ha sido intencionalmente dejada en blanco.

# Apéndice B

## Scripts

### B.1. Recolección de estadísticas

OMNeT++ permite definir las estadísticas a recolectar por los diferentes módulos en su implementación, sin embargo no siempre se quieren registrar todas ellas debido al espacio que esto requiere. En particular las medidas de vectores, donde se registra cada valor emitido por el modulo en función del tiempo para la totalidad de la simulación, crecen rápidamente al aumentar la cantidad de elementos en la simulación como pueden ser los usuarios. Se tomó la decisión de registrar todas las estadísticas numéricas, como promedios o valores máximos y mínimos, ya que el tamaño de estas es aceptable. Mientras que para los vectores se registraron únicamente aquellos de interés, esto se especifica en la configuración de la simulación de la siguiente forma

```
**splitManager**.outgoingDataRate:vector.vector-recording
    = true
**splitManager**.queueingTime:vector.vector-recording =
    true
**splitManager**.DataRate:vector.vector-recording = true
**splitManager**.queueLength:vector.vector-recording =
    true
```

La sintaxis de los símbolos `**` implica que se aplicará para los modulos que contengan cualquier sentencia de caracteres en dicho lugar, con esto se habilitan las métricas para todos los equipos con el modulo `splitManager`, es decir los DU y CU.

### B.2. `get_results.py`

Este script permite correr las simulaciones básicas, consistentes de usuarios de un solo tipo de tráfico, y extraer los resultados. Para correrlo, se debe ejecutar el siguiente comando:

```
python get_results.py -c <config> -r <filter> -f <ini file>
```

## Apéndice B. Scripts

La opción `-c` permite seleccionar la configuración a utilizar, utilizada por ejemplo para seleccionar los tipos de tráfico a simular. La opción `-r` permite filtrar las simulaciones a ejecutar de todas las combinaciones de parámetros implementadas para la configuración seleccionada. Se puede seleccionar las simulaciones con un valor específico de un parámetro, por ejemplo el punto de *split* a utilizar es definido con la variable “splitPoint”, si se quisiera ejecutar únicamente las simulaciones con la opción 4 de *split* se debería incluir el filtro `-r ‘$splitPoint=4’`. Si no se especifica ningún filtro, se ejecutarán todas las simulaciones. Por último, la opción `-f` permite seleccionar el archivo de configuración a utilizar, el cual debe ser un archivo de configuración de OMNeT++. Este recolecta las estadísticas configuradas en los archivos `.ini`, las procesa utilizando una herramienta provista por OMNeT++ para almacenarlas en un archivo CSV con el que se puede trabajar de manera cómoda en Python. En general las estadísticas de todos los conjuntos de parámetros y repeticiones asociados a una misma configuración se procesan a un mismo archivo. Los vectores sin embargo son almacenados cada uno en su propio archivo debido a limitaciones de la herramienta, además de evitar generar archivos demasiado grandes.

### B.3. run\_mixed.py

El lenguaje de especificación de las simulaciones de OMNeT++ no permite la asignación de subconjuntos de los vectores de módulos en base a parámetros. Esto dificulta la implementación de simulaciones mixtas con una cantidad de usuarios paramétrica, dado que a cada subconjunto del vector de usuarios se le debe asignar un tipo de tráfico distinto. Para solucionar este problema, se implementó un script que crea un archivo de simulación sustituyendo los valores de los parámetros en un archivo de configuración de OMNeT++ partiendo de un archivo en forma de plantilla que contiene los parámetros a sustituir. Para correrlo, se debe ejecutar el siguiente comando:

```
python run_mixed.py -r <filter> -f <ini file>
```

La opción `-r` permite filtrar las simulaciones a ejecutar de forma similar a `get_results.py`. La opción `-f` permite seleccionar el archivo de configuración a utilizar como plantilla para la simulación, el cual debe ser un archivo de configuración de OMNeT++. Las estadísticas son procesadas de igual forma que el caso anterior.

### B.4. Analisis de resultados

Para analizar los resultados se generaron diferentes archivos utilizando *notebooks* de Python para cada simulación. En estos se leen las estadísticas recolectadas por los scripts anteriores y se grafican las diferentes medidas de interés.

# Apéndice C

## Medidas de *performance*

### C.1. Ambiente de Simulación

Las simulaciones fueron realizadas en una sistema con los siguientes recursos:

Procesador	Intel Core i5-12600K @ 3.7 GHz
RAM	32 GB
SO	Ubuntu 24.04.01

### C.2. Tiempo de ejecución de las simulaciones

A continuación se incluyen los tiempos necesarios para correr la simulación de cada experimento. En todos los casos el tiempo simulado fueron 60 segundos.

Topología	Experimento	Tiempo total de ejecución
Topología 0	Simu5G	1 m 46 s
Topología 0	SimuSplit	1m 55 s
Topología 1	Comparación de cantidad de usuarios	39 m 56 s
Topología 1	Interferencia de otras radiobases	48 m 13 s
Topología 1	Variación del tamaño del paquete	78 m 35 s
Topología 2	Comparación de distribución de usuarios por celda	4m 20 s
Topología 3	Comparación de cantidad de usuarios	8m 35 s

Esta página ha sido intencionalmente dejada en blanco.

# Apéndice D

## Gráficas

Aquí se presentan algunas gráficas obtenidas en base a las simulaciones realizadas sobre las distintas topologías donde las observaciones a realizar ya fueron mencionadas en el capítulo 6 en base a otros casos con el mismo comportamiento como puede ser un típico de tráfico distinto. Las Figuras D.1 y D.2 corresponden a la topología 1 con el experimento en el cual se comparan distintas cantidades de usuarios utilizando tráfico de tipo *burst*. En las Figuras D.3 y D.4 se ilustra el *delay* en la topología 1 con una comparación entre la cantidad de usuarios para el caso en el que estos se encuentran posicionados de manera aleatoria en vez de a una misma distancia de la DU. Finalmente las Figuras D.5 y D.6 se observa el experimento en el que se comparan distintos tamaños de paquete en la topología 1 con tráfico *burst*.

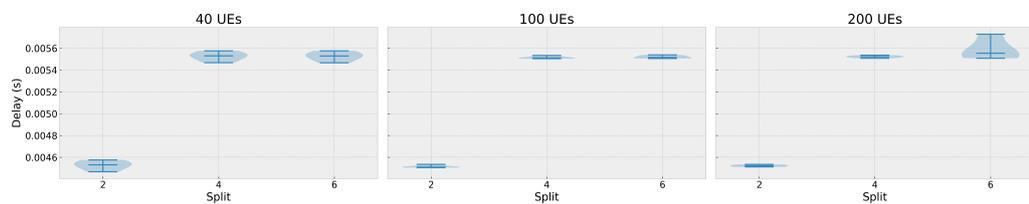


Figura D.1: Topología 1: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de tráfico *burst* en sentido DL promediadas sobre todos los usuarios. Cada columna corresponde a una simulación con una determinada cantidad de usuarios. Se comparan las tres opciones de *delay* implementadas.

## Apéndice D. Gráficas

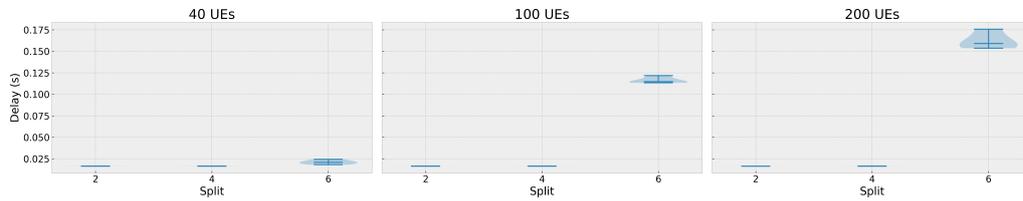


Figura D.2: Topología 1: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de tráfico *burst* en sentido UL promediadas sobre todos los usuarios. Cada columna corresponde a una simulación con una determinada cantidad de usuarios. Se comparan las tres opciones de *delay* implementadas.

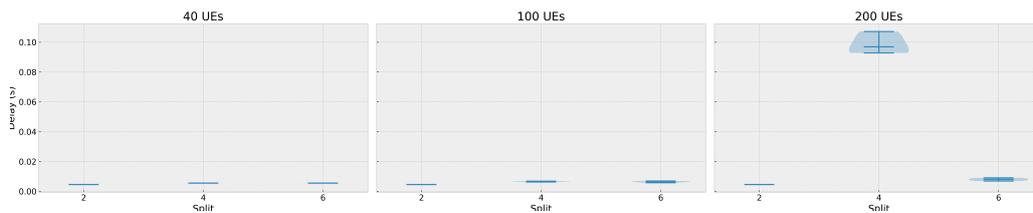


Figura D.3: Topología 1: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de tráfico VoIP en sentido DL promediadas sobre todos los usuarios donde estos son ubicados de manera aleatoria. Cada columna corresponde a una simulación con una determinada cantidad de usuarios y se comparan las tres opciones de *split* implementadas.

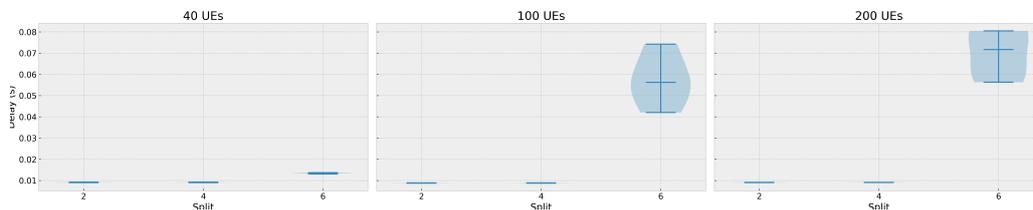


Figura D.4: Topología 1: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de tráfico VoIP en sentido UL promediadas sobre todos los usuarios donde estos son ubicados de manera aleatoria. Cada columna corresponde a una simulación con una determinada cantidad de usuarios y se comparan las tres opciones de *split* implementadas.

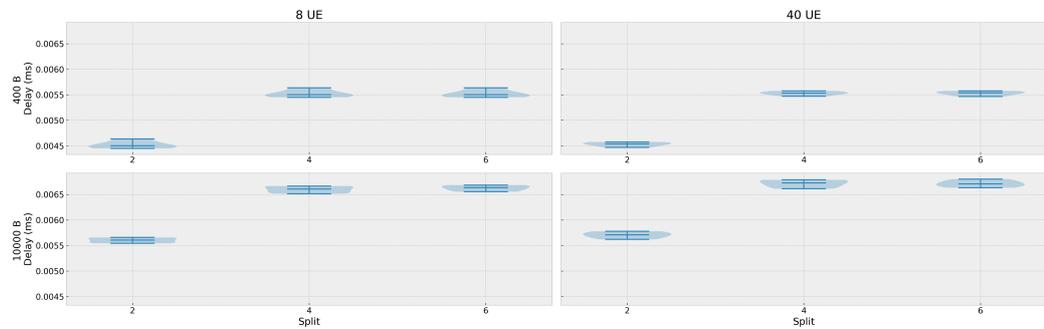


Figura D.5: Topología 1: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de tráfico *burst* en sentido DL promediadas sobre todos los usuarios. Cada columna corresponde a una simulación con una determinada cantidad de usuarios, mientras que las filas corresponden a un tamaño de paquete generado distinto. Se comparan las tres opciones de *delay* implementadas.

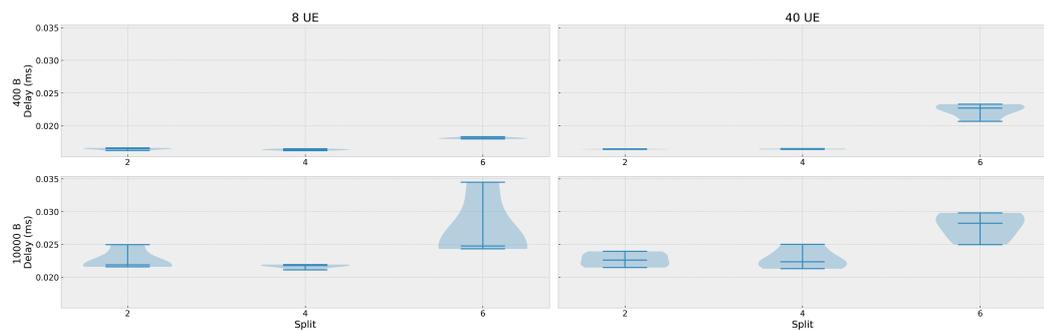


Figura D.6: Topología 1: Gráfica de violín para las métricas de *delay* extremo a extremo en una simulación con usuarios de tráfico *burst* en sentido UL promediadas sobre todos los usuarios. Cada columna corresponde a una simulación con una determinada cantidad de usuarios, mientras que las filas corresponden a un tamaño de paquete generado distinto. Se comparan las tres opciones de *delay* implementadas.

Esta página ha sido intencionalmente dejada en blanco.

# Referencias

- [1] 3GPP. Study on new radio access technology: Radio access architecture and interfaces. Technical Report TS 38.801 V14.0.0 (2017-03), 3GPP, 2017.
- [2] 3GPP. Nr; physical layer procedures for data. Technical Report TS 38.214 V18.4.0 (2024-09), 3GPP, 2024.
- [3] 3GPP. Study on channel model for frequencies from 0.5 to 100 ghz. Technical Report TS 38.901 V18.0.0 (2024-04), 3GPP, 2024.
- [4] 3GPP. System architecture for the 5g system (5gs). Technical Report TS 23.501 V19.1.0 (2024-09), 3GPP, 2024.
- [5] Fredrik Alriksson, Du Ho Kang, Chris Phillips, Jose Luis Pradas, and Ali Zaidi. XR and 5G: Extended reality at scale with time-critical communication. *Ericsson Technology Review*, 2021(8):2–13, August 2021.
- [6] Valentín Brienza and Marcos Rapetti. Simusplit: Simulación para la gestión dinámica de splits y optimización de recursos en redes 5g/6g. <https://gitlab.fing.edu.uy/mobile-optical-networks/simusplit>, 2024.
- [7] Aleksandra Checko, Henrik L. Christiansen, Ying Yan, Lara Scolari, Georgios Kardaras, Michael S. Berger, and Lars Dittmann. Cloud ran for mobile networks—a technology overview. *IEEE Communications Surveys Tutorials*, 17(1):405–426, 2015.
- [8] TE Connectivity. Mass connectivity in the 5g era. Technical report, 2018.
- [9] Marcin Dryjanski. 5g core network - a short overview, 2017. 5G\_achitecture.
- [10] NEC Corporation Ericsson AB, Huawei Technologies Co. Ltd and Nokia. *eCPRI Specification V2.0*, 2010.
- [11] Abdulhalim Fayad, Tibor Cinkler, and Jacek Rak. Toward 6g optical fronthaul: A survey on enabling technologies and research perspectives. *IEEE Communications Surveys Tutorials*, 2024.
- [12] Small Cell Forum. Options for split ran: Flexibility within a common framework. Technical Report SCF260.01, SCF, 2023.

## Referencias

- [13] Mohammad Asif Habibi, Meysam Nasimi, Bin Han, and Hans D. Schotten. A comprehensive survey of ran architectures toward 5g mobile communication system. *IEEE Access*, 7:70371–70421, 2019.
- [14] Davit Harutyunyan and Roberto Riggio. Flex5g: Flexible functional split in 5g networks. *IEEE Transactions on Network and Service Management*, PP:1–1, 07 2018.
- [15] Md. Farhad Hossain, Ayman Uddin Mahin, Topojit Debnath, Farjana Binte Mosharrof, and Khondoker Ziaul Islam. Recent research in cloud radio access network (c-ran) for 5g cellular systems - a survey. *Journal of Network and Computer Applications*, 139:31–48, 2019.
- [16] Lucas Inglés. Aprendizaje profundo para la asignación de recursos en redes 5g. Tesis de maestría, UdelaR, 2023.
- [17] International Telecommunication Union. One-way transmission time. Technical report, ITU-T, Geneva, Switzerland, 05 2003.
- [18] E. Dahlman J. Sköld, S. Parkvall. *5G NR - The Next Generation Wireless Access Technology*. Academic press, 2018.
- [19] Bahram Khan, Nidhi Nidhi, Hatem OdetAlla, Adam Flizikowski, Albena Mihovska, Jean-Frédéric Wagen, and Fernando Velez. Survey on 5g second phase ran architectures and functional splits. October 2022.
- [20] M. Khan, R.S. Alhumaima, and H.S. Al-Raweshidy. Reducing energy consumption by dynamic resource allocation in c-ran. In *2015 European Conference on Networks and Communications (EuCNC)*, pages 169–174, 2015.
- [21] V. López L. Velasco. *Elastic Optical Networks*. Springer, 2016.
- [22] Line M. P. Larsen, Aleksandra Checko, and Henrik L. Christiansen. A survey of the functional splits proposed for 5g mobile crosshaul networks. *IEEE Communications Surveys Tutorials*, 21(1):146–172, 2019.
- [23] Alberto Martinez Alba and Wolfgang Kellerer. A dynamic functional split in 5g radio access networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.
- [24] MathWorks. 5G Toolbox, 2024. MATLAB Software Package.
- [25] Jon Matias, Jokin Garay, Nerea Toledo, Juanjo Unzilla, and Eduardo Jacob. Toward an sdn-enabled nfv architecture. *IEEE Communications Magazine*, 53(4):187–193, 2015.
- [26] Salma Matoussi, Ilhem Fajjari, Nadjib Aitsaadi, and Rami Langar. User slicing scheme with functional split selection in 5g cloud-ran. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–8, 2020.

- [27] Salma Matoussi, Ilhem Fajjari, Salvatore Costanzo, Nadjib Aitsaadi, and Rami Langar. 5g ran: Functional split orchestration optimization. *IEEE Journal on Selected Areas in Communications*, 38(7):1448–1463, 2020.
- [28] Sourav Mondal and Marco Ruffini. Optical front/mid-haul with open access-edge server deployment framework for sliced o-ran. *IEEE Transactions on Network and Service Management*, 19(3):3202–3219, 2022.
- [29] Meroua Moussaoui, Emmanuel Bertin, and Noel Crespi. Telecom business models for beyond 5g and 6g networks: Towards disaggregation? In *2022 1st International Conference on 6G Networking (6GNet)*, pages 1–8, 2022.
- [30] Martin Klaus Müller, Fjolla Ademaj, Thomas Dittrich, Agnes Fastenbauer, Blanca Ramos Elbal, Armand Nabavi, Lukas Nagel, Stefan Schwarz, and Markus Rupp. Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator. *EURASIP Journal on Wireless Communications and Networking*, 2018(1):17, September 2018.
- [31] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Viridis. Simu5g—an omnet++ library for end-to-end performance evaluation of 5g networks. *IEEE Access*, 8:181176–181191, 2020.
- [32] Navid Nikaein. Processing radio access network functions in the cloud: Critical issues and modeling. In *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services, MCS '15*, page 36–43, New York, NY, USA, 2015. Association for Computing Machinery.
- [33] Stefan Parkvall, Erik Dahlman, Anders Furuskar, and Mattias Frenne. Nr: The new 5g radio access technology. *IEEE Communications Standards Magazine*, 1(4):24–30, 2017.
- [34] Natale Patriciello, Sandra Lagén, Biljana Bojovic, and Lorenza Giupponi. An E2E simulator for 5g NR networks. *CoRR*, abs/1911.05534, 2019.
- [35] Utpal Paul, Anand Prabhu Subramanian, Milind Madhav Buddhikot, and Samir R. Das. Understanding traffic dynamics in cellular data networks. In *2011 Proceedings IEEE INFOCOM*, pages 882–890, 2011.
- [36] Michele Polese, Leonardo Bonati, Salvatore D’Oro, Stefano Basagni, and Tommaso Melodia. Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys Tutorials*, 25(2):1376–1411, 2023.
- [37] Stefan Pratschner, Bashar Tahir, Ljiljana Marijanovic, Mariam Mussbah, Kiril Kirev, Ronald Nissel, Stefan Schwarz, and Markus Rupp. Versatile mobile communications simulation: the Vienna 5G Link Level Simulator. *EURASIP Journal on Wireless Communications and Networking*, 2018(1):226, September 2018.

## Referencias

- [38] B. J. Puttnam, M. van den Hout, G. Di Sciullo, R. S. Luis, G. Rademacher, J. Sakaguchi, C. Antonelli, C. Okonkwo, and H. Furukawa. 22.9 pb/s data-rate by extreme space-wavelength multiplexing. In *49th European Conference on Optical Communications (ECOC 2023)*, volume 2023, 2023.
- [39] Jordi Pérez-Romero, Oriol Sallent, Antoni Gelonch, Xavier Gelabert, Bleron Klaiqi, Marcus Kahn, and David Campoy. A tutorial on the characterisation and modelling of low layer functional splits for flexible radio access networks in 5g and beyond. *IEEE Communications Surveys Tutorials*, 25(4):2791–2833, 2023.
- [40] Micaela Rodriguez, Máximo Pirri, Lucas Inglés, Claudina Rattaro, and Alberto Castro. Integrating optical and mobile networks: A comprehensive end-to-end simulation platform. In *2024 24th International Conference on Transparent Optical Networks (ICTON)*, 2024.
- [41] M. Rodríguez. Herramienta de simulación de una red end-to-end óptico-móvil, 2024.
- [42] Syed Hashim Ali Shah, M. Kaleem Iqbal, and Tassadaq Hussain. Comparison between wimax and 3gpp lte, 2009.
- [43] Sheldon. Wdm basics: Understanding wavelength division multiplexing technology, 2020.
- [44] João Soares, Carlos Gonçalves, Bruno Parreira, Paulo Tavares, Jorge Carapinha, João Paulo Barraca, Rui L. Aguiar, and Susana Sargento. Toward a telco cloud environment for service functions. *IEEE Communications Magazine*, 53(2):98–106, 2015.
- [45] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Simutools '08*, Brussels, BEL, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [46] Huajun Wang, M.M. Aftab Hossain, and Cicek Cavdar. Cloud ran architectures with optical and mm-wave transport technologies. In *2017 19th International Conference on Transparent Optical Networks (ICTON)*, 2017.
- [47] Jun Wu, Zhifeng Zhang, Yu Hong, and Yonggang Wen. Cloud radio access network (c-ran): a primer. *IEEE Network*, 29(1):35–41, 2015.
- [48] Ming Yan, Chien Aun Chan, Wenwen Li, Ling Lei, Qianjun Shuai, Andre F. Gygax, and I. Chih-Lin. Assessing the energy consumption of 5g wireless edge caching. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019.



Esta es la última página.  
Compilado el viernes 17 enero, 2025.