



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



A5iGnator: Framework para la Implementación de Algoritmos de Asignación de Recursos en 5G

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

Alexis Muzante, Martin Rizzo, Enzo Davyt

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

TUTORES

Pablo Belzarena..... Facultad de Ingeniería, UdelaR
Claudina Rattaro Facultad de Ingeniería, UdelaR

TRIBUNAL

Germán Capdehourat Facultad de Ingeniería, UdelaR
Federico La Rocca Facultad de Ingeniería, UdelaR
Bruno Benedetti Facultad de Ingeniería, UdelaR
Gabriela Pereyra ANTEL, Uruguay

Montevideo
lunes 1 febrero, 2021

A5iGnator: Framework para la Implementación de Algoritmos de Asignación de Recursos en 5G, Alexis Muzante, Martin Rizzo, Enzo Davyt.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 156 páginas.
Compilada el lunes 1 febrero, 2021.
<http://iie.fing.edu.uy/>

Agradecimientos

A los tutores Claudina y Pablo que brindaron apoyo y acompañaron durante todo el proyecto.

A la Universidad Técnica de Vienna (TU Wien) por brindarnos una licencia para acceder al software Vienna 5G System Level Simulator.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

Debido a la incipiente instauración de las tecnologías 5G NR (New Radio), quinta generación de tecnologías de telefonía móvil, ha surgido la oportunidad de profundizar los conocimientos en este campo de investigación. Esta tecnología se encuentra aún en etapas de estandarizaciones y definiciones, la cual deja libre ciertos aspectos que cada fabricante o grupo investigador se encargará de llenar proponiendo la solución más adecuada. Uno de estos aspectos es el scheduling (algoritmo implementado para repartir los recursos disponibles).

Un primer objetivo del proyecto es el estudio de la tecnología 5G en cuanto a su arquitectura general por lo tanto se exponen las mejoras introducidas respecto a LTE en capa 1 y capa 2. Posteriormente se hace foco en la subcapa MAC para estudiar distintas opciones de algoritmos de scheduling que pueden ser implementados en la nueva generación de telefonía.

Un segundo pilar del proyecto es el estudio del funcionamiento del Vienna 5G SLS (System Level Simulator), un simulador de redes 5G del cual posee la licencia la Facultad de Ingeniería. Se explica parte del código fuente del simulador, se analizan las diferentes funcionalidades que posee y cómo se relacionan con la tecnología 5G. Mediante este software es posible simular algunos de los escenarios típicos de 5G, aunque el simulador posee algunas carencias que no permiten explotar a pleno los avances introducidos por la tecnología. A pesar de ello es posible probar distintos algoritmos de scheduling que ya estaban implementados en el simulador como son el Round Robin y el Best CQI (Channel Quality Indicator). También se implementan nuevos algoritmos como es el caso del Proportional Fair y una adaptación del Proportional Fair funcionando con puncturing para atender usuarios URLLC (Ultra-Reliable Low Latency Communications).

Finalmente, el objetivo principal del proyecto es la implementación de un Framework o conjunto de herramientas que permitan la simulación de algoritmos abstrayéndose del funcionamiento interno del software del simulador. El Framework permite implementar algoritmos de scheduling (basados en métrica) sobre el simulador. Funciona con una lógica de capas que aplican operaciones algebraicas a diferentes parámetros que describen el comportamiento de los usuarios en la red (CQI del usuario en el recurso a asignar, cantidad de datos enviada por el usuario, tamaño de la cola de datos). Con esta lógica de capas se logra tener independencia del código fuente del simulador y a partir de un archivo ordenado (en la implementación realizada se utiliza un archivo .mat) que contiene los ajustes del Framework, se obtiene la métrica en la que se basa el algoritmo.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos	I
Resumen	III
1. Introducción	1
1.1. Contexto y Motivación	1
1.2. Objetivos	2
1.2.1. Sistema	3
1.2.2. Uso esperado	3
1.2.3. Requerimientos conceptuales	3
1.2.4. Alcance	3
1.3. Contribuciones	4
1.4. Estructura del documento	4
2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)	7
2.1. Evolución de LTE y últimas versiones	7
2.2. Telefonía Móvil de Quinta Generación NR (5G)	9
2.2.1. Casos de uso de 5G NR	10
2.2.2. Espectro para 5G	11
2.2.3. Esquema de transmisión y estructura de la trama	12
2.2.4. HARQ, esquema de duplexación y esquemas de modulación	14
2.2.5. Canales físicos y señales de sincronización	15
2.2.6. Convivencia entre servicios eMBB y URLLC	16
2.2.7. Beamforming	18
2.2.8. MIMO masivo	19
2.3. Resumen del Capítulo	19
3. Scheduling: Asignación de recursos	21
3.1. Orígenes	21
3.2. Criterios de justicia	23
3.3. Observaciones	26
3.4. Algunos algoritmos	28
3.5. Resumen del Capítulo	32

Tabla de contenidos

4. Simulador Vienna 5G SLS	33
4.1. Estructura del simulador	34
4.1.1. Inicialización	34
4.1.2. Pregeneración	37
4.1.3. Bucle de Simulación Principal	38
4.1.4. Post procesamiento	42
4.2. Limitantes a destacar del Simulador Vienna 5G SLS	42
4.3. Resumen del Capítulo	43
5. Modelo de tráfico	45
5.1. Modelo de tráfico requerido por los usuarios	45
5.1.1. Adaptación del simulador al modelo de tráfico	46
5.2. Validación del modelo de tráfico	49
5.2.1. Verificación de traspaso de déficit	51
5.2.2. Verificación de asignación mínima	52
5.3. Resumen del Capítulo	53
6. Proportional Fair Generalizado	55
6.1. Diseño del algoritmo	55
6.2. Implementación en simulador Vienna 5G SLS	57
6.3. Validación del algoritmo implementado	60
6.4. Resumen del Capítulo	68
7. Grilla con minislots y Proportional Fair con puncturing	69
7.1. Grilla con minislots	69
7.1.1. Implementación grilla NR y grilla con minislots	69
7.1.2. Validación grilla NR y NR_minislot	72
7.2. Proportional Fair con Puncturing	77
7.2.1. Agregado de usuarios URLLC	77
7.2.2. Implementación de puncturing en el simulador	77
7.2.3. Validación Proportional Fair con Puncturing	79
7.3. Resumen del Capítulo	89
8. Framework	91
8.1. Modelado de scheduler	91
8.1.1. Combinación lineal	92
8.1.2. Producto con exponentes de peso	94
8.1.3. Prefunciones	94
8.1.4. Desempate	95
8.1.5. Modelado de ajustes	95
8.1.6. Ejemplo de algoritmo	96
8.2. Implementación sobre el simulador Vienna 5G SLS	99
8.3. Validación del Framework	101
8.4. Ejemplo de uso del Framework	105
8.5. Resumen del Capítulo	108

9. Conclusiones y trabajo futuro	109
9.1. Conocimientos adquiridos	109
9.2. Valor agregado introducido por el proyecto	110
9.3. Trabajo Futuro	111
A. Duplexación y acceso al medio	113
A.1. FDD y TDD	113
A.2. Técnicas de acceso al medio	113
A.2.1. FDMA (Frequency Division Multiple Access)	113
A.2.2. TDMA (Time Division Multiple Access)	113
A.2.3. CDMA (Code Division Multiple Access)	113
B. Red de telecomunicaciones Celular	115
B.1. Estructura de una red de telecomunicaciones	115
B.1.1. Núcleo de red	115
B.1.2. Red de transporte, distribución y agregación	115
B.1.3. Red de acceso	115
B.2. Outdoor de una radiobase	116
B.3. Clasificación de celdas	116
B.3.1. Según cantidad de sectores	116
B.3.2. Según su alcance, potencia de transmisión y área de cobertura	116
B.4. Evolución de la telefonía celular	116
B.4.1. Telefonía Móvil de Primera Generación AMPS (1G)	116
B.4.2. Telefonía Móvil de Segunda Generación GSM (2G)	117
B.4.3. Telefonía Móvil de Tercera Generación UMTS (3G)	117
B.4.4. Telefonía Móvil de Cuarta Generación LTE (4G)	118
B.4.5. Descripción general de la primer versión	118
C. Entes reguladores, organizaciones de estándares y foros industria-	
les	121
D. Tablas CQI para LTE y 5G	125
E. Uso del Framework en simulador Vienna	127
Referencias	129
Glosario de Siglas	135
Índice de tablas	139
Índice de figuras	140

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 1

Introducción

1.1. Contexto y Motivación

El presente trabajo se centra en la tecnología de red celular de quinta generación 5G NR (New Radio) desarrollada por la 3GPP (3rd Generation Partnership Project [1]) la cual promete ampliar y mejorar de forma sustancial los servicios que se asocian a telefonía móvil. Este proyecto aborda, específicamente, el problema de la asignación de recursos (o scheduling) en la red 5G.

En la década del 80 “la telefonía móvil consistía en ofrecer un acceso “vía radio” a un abonado de telefonía de tal forma que pudiera realizar y recibir llamadas dentro del radio de cobertura del sistema” [2]. En esos años se comenzaban a hacer comerciales los primeros sistemas de telefonía móvil. El concepto mutó hasta convertirse en lo que es en la actualidad: la telefonía móvil se entiende como un vehículo de comunicación permanente que no solamente se utiliza para la interacción entre personas sino que también abrió un abanico de posibilidades que ampara desde el entretenimiento y ocio hasta la modificación de la forma de trabajar. El fenómeno de la introducción de los smartphones en el mercado (junto con las aplicaciones que implementan) sumado a la universalización de la posibilidad de acceder a ellos provocó que la red móvil debiera evolucionar.

En la última década los requerimientos de ancho de banda y velocidad de transmisión de datos se volvieron muy demandantes lo que ha obligado a los fabricantes a mejorar sus productos y a los organismos de regulación internacional a trabajar en estándares más competitivos, al punto de exigir a la red móvil velocidades de respuesta similares a la red cableada, promesa que la red 5G pretende cumplir. Los servicios [3] de banda súper ancha eMBB (Enhanced Mobile Broadband) que ofrece 5G atienden los problemas y exigencias naturales del crecimiento de la red telefónica tradicional. Además de aportar soluciones en estos servicios, 5G propone mejorar los servicios de mMTC (Massive Machine Type Communications) que tienen su auge en la actualidad con todo lo relacionado a IoT (Internet of Things, o Internet de las cosas en español), donde el servicio se centra en millones de dispositivos enviando muchos paquetes pequeños de información. También prevé abordar problemas relacionados con URRLC (Ultra Reliable Low Latency

Capítulo 1. Introducción

Communications) que se trata de servicios que requieren muy baja latencia y alta fiabilidad, como pueden ser transferencias bancarias, o casos más ambiciosos como operaciones médicas a larga distancia o ciudades inteligentes. Los servicios que requieren URRLC son prácticamente una novedad en el mundo de las telecomunicaciones, por lo que su implementación se incorpora recién en las redes 5G NR.

En las nuevas redes 5G NR se introducen algunas novedades como la estructura variable de la trama [4], operación en ondas milimétricas [3] y el network slicing [5] [6].

También se incorpora el concepto de puncturing [7] [8] que es un mecanismo que prioriza usuarios URLLC por sobre eMBB (basado en que los usuarios URLLC pueden interrumpir a los usuarios eMBB para lograr baja latencia).

Dados los diversos servicios ofrecidos, así como la variedad de nuevas tecnologías incorporadas, se abren nuevos campos para explorar. Entre ellos se encuentra la reformulación de los algoritmos de asignación de recursos (scheduling) realizada por las radiobases. Esta asignación debe permitir la convivencia entre los distintos servicios ofrecidos y lograr un mejor aprovechamiento de los recursos teniendo en cuenta las nuevas características.

La etapa de simulación es de gran relevancia en el proceso de diseño. En el caso de los schedulers esta etapa se ve enlentecida por la necesidad de comprender el funcionamiento de cada simulador a utilizar (para poder incorporar la propuesta dentro del código del simulador). Por esto nace la necesidad de desarrollar una herramienta que agilice la incorporación de este tipo de algoritmos a los simuladores y de esta forma brindar un entorno de trabajo intuitivo y estándar abstrayendo al investigador de la profundidad del código del simulador. La motivación del proyecto reside en la creación de dicha herramienta, para lo cual se deben investigar distintos algoritmos como un medio para comprender y modelar un sistema de scheduling estándar. Modelar la asignación de recursos de esta forma permite la independencia del Framework sobre los distintos simuladores, lo cual es una ventaja.

En los últimos años se han desarrollado distintos simuladores para redes 5G [9] [10] [11] [12]. Con el objetivo de mostrar el funcionamiento se implementan distintos algoritmos así como el Framework diseñado sobre el simulador “Vienna 5G System Level Simulator” [13]. Este simulador fue creado en la Universidad Tecnológica de Viena (versión 1.0 liberada en 2018). Es un simulador de sistema (ideal para un análisis de capacidad y de algoritmos de scheduling) que funciona en Matlab (versión 2018 o superior).

1.2. Objetivos

El objetivo principal del proyecto es crear un Framework para la generación de algoritmos de scheduling e implementarlo sobre el software Vienna 5G SLS 1.0.

1.2.1. Sistema

El sistema está compuesto por un conjunto de herramientas de software destinadas a la simulación de algoritmos de scheduling con base en el simulador Vienna 5G SLS. Se implementa mediante un sistema de superposición de capas en las cuales se realizan diferentes operaciones algebraicas. De este modo se genera la métrica (en la que se basa el algoritmo) sin tener que profundizar en el propio funcionamiento del simulador.

1.2.2. Uso esperado

El Framework está diseñado para la simulación de algoritmos de scheduling en redes 5G. Es posible ejecutarlo sobre escenarios con múltiples usuarios cuyo comportamiento puede ser respondiendo a servicios eMBB o URLLC, con la posibilidad de que convivan utilizando puncturing. Los parámetros a utilizar para la elaboración de la métrica son el CQI de cada usuario en el recurso a asignar, la cantidad de datos que ha enviado el usuario (en una ventana de tiempo) y el largo de la cola de datos que aún resta recibir cada usuario. Es posible agregar más parámetros si el usuario que utiliza el Framework lo requiere. La métrica que elabora se utiliza para la asignación de los recursos de los usuarios eMBB mientras que para los usuarios URLLC se debe seleccionar algún algoritmo de scheduling ya definido. El Framework no se encarga de modificar los parámetros del escenario en que se realiza la simulación.

1.2.3. Requerimientos conceptuales

El Framework permite configurar distintos algoritmos de scheduling a partir de parámetros y características de la red. Dichos algoritmos se basan en una métrica, a partir de la cual se asignan los recursos al usuario con mejor métrica. El Framework se ejecuta sobre el simulador Vienna 5G SLS. El sistema recibe las entradas en forma de archivo .mat el cual debe respetar un orden que se detalla en el Capítulo 8. La entrada del Framework son vectores y matrices que indican cada capa de operaciones que se utiliza, los parámetros involucrados en la métrica, los coeficientes o exponentes con los que se operará con los parámetros y también se indican funciones auxiliares que es posible utilizar (prefunctions).

1.2.4. Alcance

El alcance del proyecto abarca todo el estudio previo que es necesario para la elaboración final del Framework. Se debe desarrollar sobre el funcionamiento general de la tecnología 5G, profundizando en el conocimiento de la subcapa MAC y física (específicamente algoritmos de scheduling). Se debe estudiar el simulador Vienna 5G SLS, se debe implementar y validar un modelo de tráfico para la información requerida por los usuarios. Se deben implementar y validar un par de algoritmos sobre el simulador (Proportional Fair Generalizado y su ampliación utilizando puncturing). Y finalmente se debe implementar y validar el Framework.

1.3. Contribuciones

Un gran desafío que se presentó fue que el simulador Vienna 5G SLS se encontraba en una versión prematura en el transcurso del proyecto y no contaba con soporte en su foro. Entre las carencias del simulador se destacan:

- No tenía implementado el uso de minislots (fundamental para aplicar puncturing).
- Carecía de modelo de tráfico para los usuarios (funcionaba en full buffer).
- No tenía desarrollada la capacidad de utilizar ondas milimétricas.
- No tenía escenarios acordes a los requerimientos NR.

Partiendo de esta base se comenzó a subsanar parte de dichas carencias completando lo necesario para el proyecto. El resultado fue la implementación de un algoritmo de Proportional Fair para usuarios eMBB y su extensión con puncturing [8] para la convivencia de servicios eMBB y URLLC (para ello fue necesario adaptar la estructura de la grilla de recursos introduciendo minislots en el funcionamiento del simulador). Se implementó un modelo de tráfico, para los usuarios, que permite que las simulaciones no sean en full buffer. Se implementaron además escenarios de prueba y archivos pertinentes necesarios para realizar la simulación. Finalmente se creó un Framework que permite implementar algoritmos de scheduling basados en una métrica. La característica principal del Framework es que logra abstraerse del funcionamiento interno del simulador y a partir de un archivo .mat (con los parámetros de usuario a utilizar, operaciones definidas y un criterio de desempate) se crea la métrica en la que se basa la asignación de recursos.

1.4. Estructura del documento

El resto de esta documentación se organiza en los siguientes capítulos:

- Capítulo 2: Red 5G NR y transición con LTE (Long Term Evolution). Se describe las novedades que introduce 5G y su transición con la tecnología LTE.
- Capítulo 3: Scheduling: Asignación de recursos. Se desarrolla sobre el concepto de scheduling, su origen histórico y como cambió su planteo con la evolución de las redes de comunicaciones. Se abordan conceptos y resultados que dan una base más sólida sobre el tema, complementado con ejemplos de algoritmos y sus características.
- Capítulo 4: Simulador Vienna 5G SLS. Se describe la estructura del simulador detallando los bloques más relevantes y las modificaciones realizadas.
- Capítulo 5: Modelo de tráfico. Se describe la implementación y validación del modelo de tráfico desarrollado.

1.4. Estructura del documento

- Capítulo 6: Proportional Fair Generalizado. Se describe la implementación y validación del algoritmo Proportional Fair implementado para usuarios eMBB.
- Capítulo 7: Grilla con minislots y Proportional Fair con Puncturing. Se describe la modificación realizada a la grilla de recursos del simulador para incorporar la utilización de minislots. Se describe la implementación y validación del algoritmo Proportional Fair con Puncturing (mecanismo que prioriza usuarios URLLC por sobre eMBB).
- Capítulo 8: Framework. Se describe la implementación, validación y ejemplo de caso de uso del Framework desarrollado.
- Capítulo 9: Conclusiones y trabajo futuro.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

Como adelanto se cometan los objetivos que plantea cumplir la red 5G (en favor de la demanda creciente del mercado de negocios, industrial y de consumo en general). Los mismos se pueden resumir en los siguientes hitos: baja latencia y alta fiabilidad en las comunicaciones (1 ms de punta a punta), gran densidad de usuarios por área con capacidad de tráfico de 10 Mbps/m², pico de tasa de datos de 10 Gbps con escenarios especiales donde se llegue a 20 Gbps, continuidad del servicio con usuarios moviéndose hasta 500 km/h, densidad de conexiones de hasta 10⁶ dispositivos por km², tasas de 100 Mbps con cobertura en grandes áreas, tres veces mejor eficiencia espectral que 4G, 100 veces mejor eficiencia energética en la red y sensores con una vida útil de hasta 10 años [5].

Previo a la descripción de la red 5G NR se enumeran las características de las versiones de LTE que anteceden a esta tecnología (un resumen de la evolución de las redes móviles se encuentra en la Sección B.4).

2.1. Evolución de LTE y últimas versiones

Los releases 8 [14] y 9 [15] de la 3GPP que corresponden a las versiones 8 y 9 conforman la base de LTE. Aportan un estándar de servicio de banda ancha móvil de gran capacidad, no obstante, para cumplir nuevos requerimientos y expectativas, las versiones que les siguen proveen mejoras adicionales y funcionalidades en diferentes áreas como se puede ver en la Figura 2.1. La versión número 10 [16] marca el comienzo de la evolución de LTE. Uno de los principales objetivos de esta versión fue asegurar que la tecnología de radio acceso de LTE fuera conforme con los requerimientos avanzados de la IMT (International Mobile Telecommunications). Así el nombre LTE-A (LTE-Advanced) es usado a menudo para las versiones posteriores a la 10 inclusive. Por otro lado, complementando los requerimientos de la ITU (International Telecommunication Union), 3GPP también definió sus propios objetivos y requerimientos para LTE-A. Entre ellos, un requerimiento importante fue la retro compatibilidad. Esto significa esencialmente que

Capítulo 2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

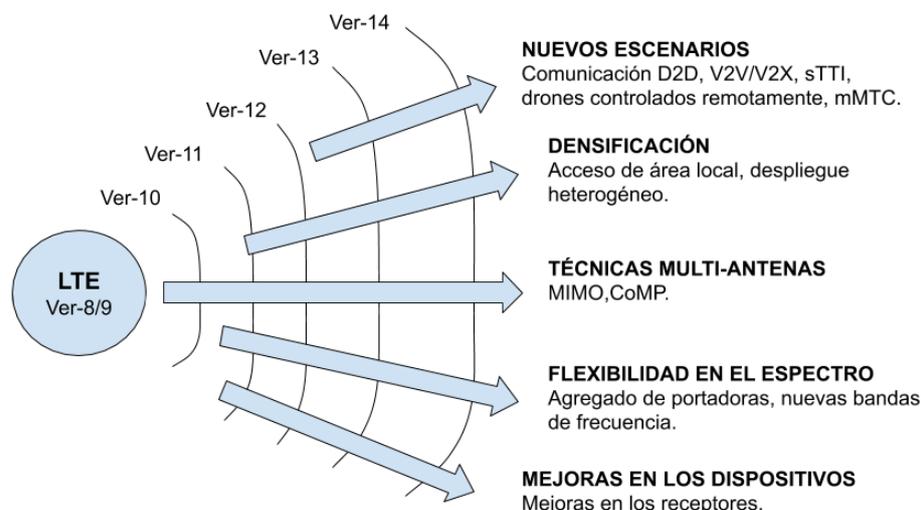


Figura 2.1: Mejoras introducidas en las versiones de LTE [17].

un dispositivo correspondiente a una versión previa de LTE debería ser capaz de acceder a un operador de la versión número 10 de LTE (aunque no sea capaz de utilizar todas las novedades de la versión 10).

La versión 10 de LTE fue terminada a finales del 2010 e introdujo flexibilidad mejorada en el espectro a través del CA (Carrier Aggregation) donde varias portadoras individuales (no necesariamente contiguas) son añadidas y utilizadas conjuntamente para la transmisión desde (o hasta) un solo dispositivo. Se pueden agregar hasta 5 portadoras diferentes (esta limitante es cambiante), las cuales no necesariamente tienen que tener el mismo ancho de banda, dando la posibilidad de tener anchos de banda de hasta 100 MHz disponibles para la transmisión de un dispositivo (ver Figura 2.2). Posteriormente introdujo transmisión multi-antena extendida, soporte para relaying^I y mejoras en torno a la coordinación de interferencia entre celdas en despliegues de redes heterogéneas^{II}. La flexibilidad espectral (en términos de soporte de ancho de banda múltiple y un diseño conjunto FDD/TDD [17]) fue considerablemente mejorada en las versiones posteriores a la versión 8, soportando el acceso a mayores anchos de banda, espectro fragmentado utilizando el agregado de portadoras y espectros de banda sin licencia utilizando LAA (License-Assisted Access).

La versión 11 [19] amplió aún más el desempeño y las capacidades de LTE; una de las características más notables de esta versión, finalizada a fines del 2012, fue la funcionalidad de radiointerfaz para la transmisión y recepción con CoMP (Coordinated Multipoint). Otro ejemplo de mejora en la versión 11 fue la incorporación

^IRelaying es una radiobase de baja potencia conectada de forma inalámbrica al resto de la red.

^{II}E. Bodanese define a las redes heterogéneas (hetnets) como redes inalámbricas que combinan múltiples tipos de nodos de acceso, como macro celdas, pico celdas y femto celdas, pudiendo interoperar entre ellos [18]

2.2. Telefonía Móvil de Quinta Generación NR (5G)

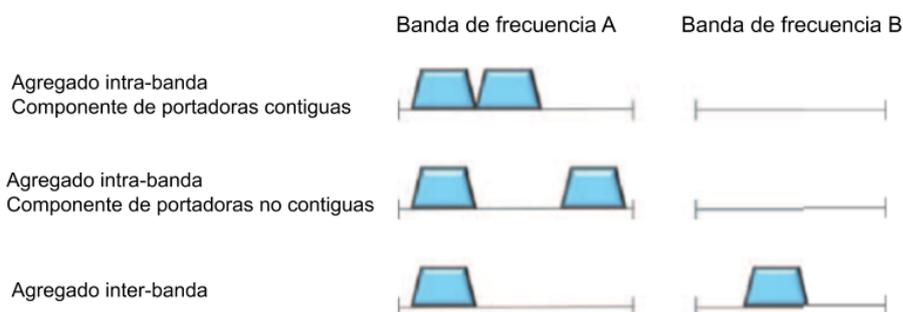


Figura 2.2: Agregado de portadoras [17].

de una nueva estructura de control de canal llamada EPDCCH (Enhanced Physical Downlink Control Channel) que introdujo información de control específica para cada dispositivo, una capacidad que puede mejorar el desempeño de beamforming en algunos escenarios. La versión 12 [20] fue terminada en 2014 y se centró en celdas pequeñas (Small Cells) y en el despliegue denso con características tales como: doble conectividad, encendido y apagado de celdas pequeñas, e incorporación de TDD (Time Division Duplex, ver Sección A.1) semi-dinámico. También permitió nuevos escenarios con la introducción de la comunicación D2D (Device-to-Device) y el aprovisionamiento de MTC (Machine Type Communication) de baja complejidad.

La versión 13 [21], finalizada al final del año 2015, marca el comienzo de LTE-Advanced Pro (a veces referida comercialmente como 4.5G) que es vista como una tecnología en el paso intermedio entre 4G y 5G NR. Incorpora el acceso asistido por licencia para admitir espectros sin licencia como complemento para los espectros con licencia, soporte mejorado para MTC y varias mejoras en agregado de portadoras, transmisión multi-antena y comunicaciones D2D. La versión 14 [22] fue terminada en el primer semestre del 2017. Además de las mejoras para algunas de las características incorporadas en versiones anteriores se introdujo soporte para comunicaciones del tipo V2V (Vehicle-to-Vehicle) y V2X (Vehicle-to-Everything), como también soporte para transmisión de área amplia con un espacio reducido entre subportadoras. La versión 15 [23] fue concluida a mediados del 2019 y la misma permitió reducir fuertemente la latencia a través de la implementación de sTTI (short Transmission Time Interval). En general esta versión extiende LTE hacia nuevos casos de uso más allá de servicios de banda ancha móvil. La versión 15 fue el centro de desarrollo para las versiones posteriores, se convirtió en parte fundamental de 5G en general y ejemplifica que LTE es parte vital en la tecnología de radio acceso de 5G.

2.2. Telefonía Móvil de Quinta Generación NR (5G)

El escenario para la nueva de generación de sistemas IMT-2020 (International Mobile Telecommunications 2020) surge en 2014 con el trabajo de la ITU-R

Capítulo 2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

(ITU Radiocommunications) WP5D (Working Party 5D [24]). El objetivo de la IMT-2020 se describe en la recomendación ITU-R M.2083 [25] (Concepción de las IMT - Marco y objetivos generales del futuro desarrollo de las IMT para 2020 en adelante). La recomendación proporciona los primeros pasos para el desarrollo de la IMT y como ésta puede servir a la sociedad, además de tratar las implicaciones espectrales que tiene la tecnología. La evaluación de dichos requerimientos culminó a mediados del 2017. Al mismo tiempo la 3GPP trabajaba en la versión 15 de LTE, que resultó ser una primer versión de NR centrada principalmente en los servicios eMBB y URRLC.

A continuación se describen los servicios que se encuentran contenidos en la red 5G. Posteriormente se comentan las diferentes características que introduce la tecnología como el espectro que maneja, el esquema de transmisión, estructura de la trama y esquema de duplexación. Se explican también los esquemas de modulación, los canales físicos, señales de sincronización y novedades como beamforming, MIMO (Multiple Input, Multiple Output) masivo y network slicing. Es necesario enfatizar el hecho que al encontrarse aún en proceso de estandarización, en el documento se mencionan los aspectos definidos en la versión 15. Puede existir la posibilidad que haya cambios agregados en algunos aspectos, por ejemplo, el uso de OFDM (Orthogonal Frequency Division Multiplexing) no es definitivo (ni será necesariamente el único mecanismo de acceso al medio), puede ser que se propongan otras formas de onda, incluso no ortogonales (pero 5G utiliza y utilizará OFDM seguro).

2.2.1. Casos de uso de 5G NR

En términos generales 5G no es específicamente una tecnología de radio acceso sino que está más precisamente definida por los casos de uso que soporta. Hasta el momento abarca tres tipos de servicios: eMBB (enhanced Mobile Broadband), mMTC (massive Machine Type Communications) y URRLC (Ultra Reliable Low Latency Communications) como se representa en la Figura 2.3.

Los servicios de eMBB son una evolución más o menos directa de los servicios de banda ancha que se ofrecen hoy con 3G, LTE y LTE-A. Cubren una amplia gama de casos, por un lado se preocupa por la cobertura de puntos críticos y áreas amplias permitiendo altas velocidades de datos y alta densidad de usuarios. También se enfoca en escenarios con baja densidad de usuarios con requerimientos de calidad de servicio superior.

Los servicios del tipo mMTC se corresponden con la inclusión en la red de dispositivos de bajo costo con una alta duración de sus baterías (varios años), tales como pueden ser sensores remotos o actuadores que monitorean equipamientos. Es un servicio emparentado con IoT que se preocupa por solucionar problemas referidos a la transmisión de información recolectada por herramientas y electrodomésticos (con la capacidad de sensar diferentes cualidades de su funcionamiento). Lo que caracteriza a estos dispositivos son el bajo consumo de energía y las bajas tasas de datos que manejan, consumen y generan una cantidad relativamente pequeña de datos por lo que el desafío está en que la red debe soportar millones de

2.2. Telefonía Móvil de Quinta Generación NR (5G)

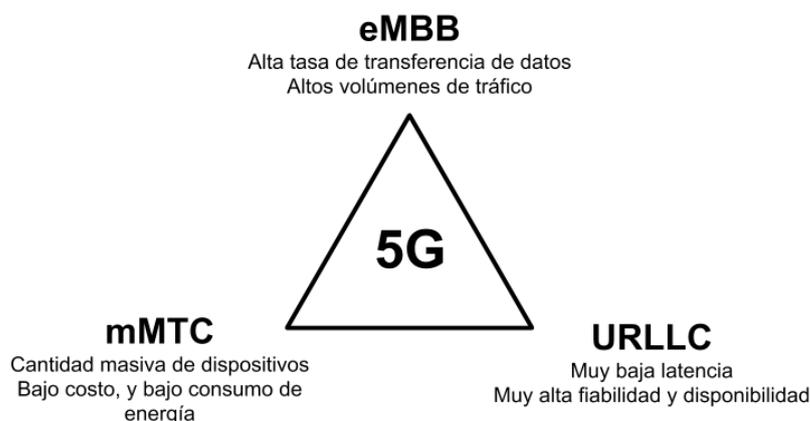


Figura 2.3: Servicios que ofrece 5G.

estos dispositivos cursando tráfico de tipo uplink.

Los servicios de la clase URLLC son los que tienen por objetivo reducir la latencia en las comunicaciones humano-máquina. Comúnmente llamada C-MTC (Critical Machine Type Communication) que requieren baja latencia, alta fiabilidad y disponibilidad. Por ejemplo se pueden mencionar las comunicaciones V2V para solucionar problemas en seguridad vial, control y automatización industrial de forma remota, al igual que cirugías médicas realizadas remotamente. En el ámbito de los videojuegos 3D también se requiere muy baja latencia (para afrontar las altas tasa de datos que se necesitan), al igual que en temas relacionados con el “internet táctil”^{III}.

2.2.2. Espectro para 5G

Para satisfacer la exigencia de los requerimientos que imponen los escenarios eMBB, como son velocidades altas de datos y altas capacidades en entornos densos, se considera usar bandas por encima de 24 GHz. Específicamente para cumplir con velocidades extremas y áreas localizadas con demanda de capacidades de tráfico muy altas se considera el despliegue utilizando frecuencias incluso por encima de 60 GHz. Estas bandas se conocen como “bandas de ondas milimétricas”, haciendo referencia a la longitud de onda.

El espectro de interés puede ser dividido en tres bandas de frecuencias: bajas, medias y altas. Las frecuencias bajas se corresponden con bandas de LTE existentes por debajo de los 2 GHz, las bandas de mayor interés aquí son las de 600 y 700 MHz. Al no ser bandas muy anchas, se estima que se utilizarán canales con un ancho máximo de 20 MHz. Las bandas de frecuencias medias están en el rango de 3 a 6 GHz y pueden proporcionar cobertura, capacidad, así como altas velocidades

^{III}El Internet Táctil será la próxima evolución del IoT, que abarcará la interacción entre persona-máquina y máquina-máquina. Permitirá sistemas interactivos en tiempo real con una serie de usos industrial, comercial y social que requieran una latencia de 1 milisegundo de punta a punta [26].

Capítulo 2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

Tabla 2.1: Estructura de la numerología y correspondiente máximo ancho de banda para los canales en 5G [27].

Frecuency Range (FR)	Δf (kHz)	T_{cp} (μs)	Slot Duration (ms)	Max. BW (MHz)
FR-1	15	4.76	1	50
	30	2.38	0.5	100
	60	1.19 4.17	0.25	100
FR-2	120	1.19 4.17	0.25	200
	240	0.6	0.125	400

de datos en canales con ancho de banda de hasta 100 MHz. El mayor interés a nivel mundial está en el rango 3300 - 4200 MHz. Las bandas de alta frecuencia están en el rango de las ondas milimétricas, las mismas se ubican por encima de los 24 GHz y permiten anchos de banda de canal hasta 400MHz (con posibilidad de anchos mayores a través del agregado de portadoras). Dichas bandas son adecuadas para la cobertura de puntos de acceso con una capacidad local y pueden proporcionar tasas de datos muy altas. El mayor interés está en el rango 24.25 - 29.5 GHz. En las bandas correspondientes a las ondas milimétricas tanto los dispositivos como las radio bases implementarán la nueva tecnología y habrá un uso más extendido de MIMO masivo.

En la versión 15 de 3GPP las bandas de frecuencia se encuentran divididas en dos rangos [3] (como se muestra en la Tabla 2.1): RF1 (Rango de Frecuencia 1) que incluye todas las bandas existentes por debajo de 6 GHz y RF2 (Rango de Frecuencia 2) que incluye bandas en el rango desde los 24.25 GHz hasta encima de los 60 GHz. Si bien la porción de espectro entre RF1 y RF2 aún no está identificada, la tecnología y especificaciones de NR pueden extenderse fácilmente.

2.2.3. Esquema de transmisión y estructura de la trama

A diferencia de lo que ocurre en LTE donde se utiliza OFDMA (Orthogonal Frequency-Division Multiple Access) para el downlink y SC-FDMA (Single Carrier Frequency Division Multiple Access) para el uplink, en NR se utiliza OFDM^{IV} como esquema de transmisión tanto en el enlace de subida como en el de bajada. Esto permite estructuras de recepción más simples y multiplexación espacial. La numerología OFDM que admite NR permite espaciados flexibles de subportadora (Δf) que van desde 15 kHz hasta 240 kHz con un cambio proporcional en la duración del prefijo cíclico (T_{cp}), lo que puede observarse en la Tabla 2.1. En el caso de LTE se utiliza un único espaciado entre subportadoras de 15 kHz con prefijo cíclico de 4,7 μs duración [17].

Se mantiene la estructura del PRB (Physical Resource Block), análogamente RB (Resource Block), de LTE con 12 subportadoras por RB, mientras que cambia la duración de símbolo y el espaciamiento entre subportadoras. En todas las variantes se mantiene la misma cantidad de REs (Resources Elements): 168 por

^{IV}OFDM es lo definido hasta el momento, pero no se descarta la posibilidad de que posteriormente se introduzcan otras formas de onda (incluso no ortogonales para servicios mMTC)

2.2. Telefonía Móvil de Quinta Generación NR (5G)

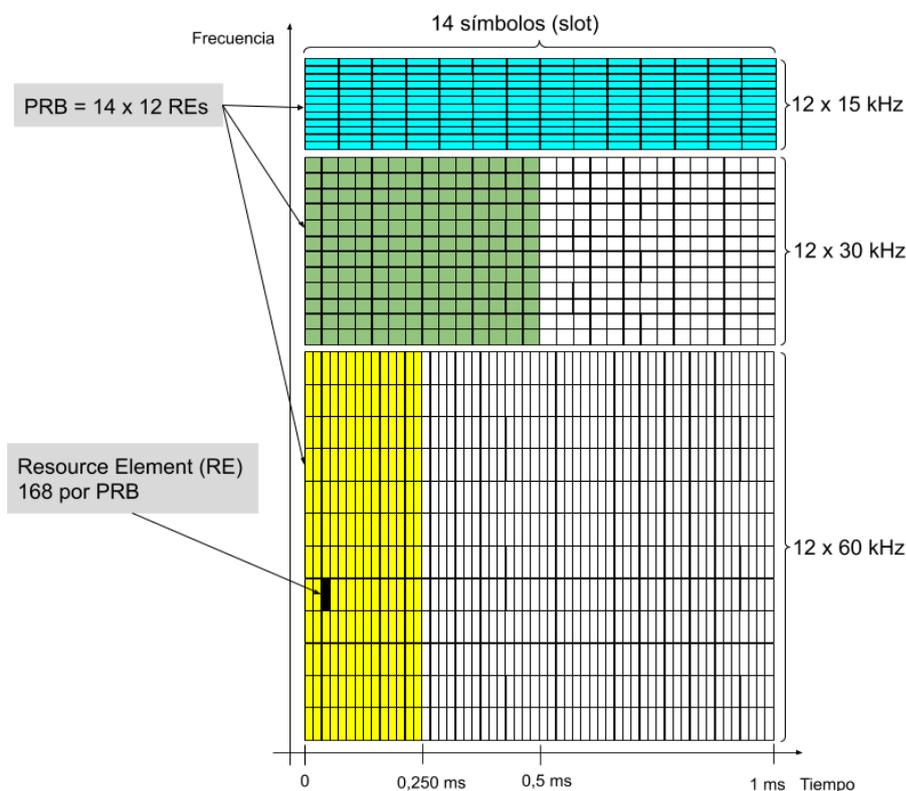


Figura 2.4: Estructura de un PRB para diferentes espaciamentos entre portadoras [29].

RB. Cada RE ocupa una subportadora en frecuencia y un símbolo en el tiempo ($12 \times 14 = 168$) como se muestra en la Figura 2.4. De esta manera se admite una amplia gama de escenarios de implementación que incluyen desde celdas grandes con subportadoras por debajo de 1 GHz hasta ondas milimétricas con amplias asignaciones de espectro. En la Figura 2.4 se observan tres numerologías diferentes (diferenciadas por la separación entre subportadoras): 15 kHz (arriba), con un tiempo de slot de 1 ms, 30 kHz (centro) con un tiempo de slot de 0,5 ms, y 60 kHz (abajo) con un tiempo de slot de 0,25 ms. La capa MAC del scheduling podría llegar a elegir libremente la portadora a la cual asigna a los usuarios decidiendo numerología y tamaño de TTI (Transmission Time Interval) [28].

Se utilizan hasta 3300 subportadoras y el máximo ancho de banda está limitado a 400 MHz, lo que resulta en el ancho de banda máximo de la portadora de 50/100/200/400 MHz para espacios de subportadora de 15/30/60/120 kHz respectivamente [4]. En el dominio del tiempo la trama (frame) NR es de 10 ms y está dividida en 10 subframes de 1 ms al igual que LTE. Cada subframe está dividido en slots de 14 símbolos OFDM y como se mencionó anteriormente, la duración del slot depende de la numerología (ver Figura 2.5). NR permite la transmisión de datos en porciones menores al slot, comúnmente denominadas minislots, funcionalidad que favorece a servicios que requieren baja latencia. De esta manera es

Capítulo 2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

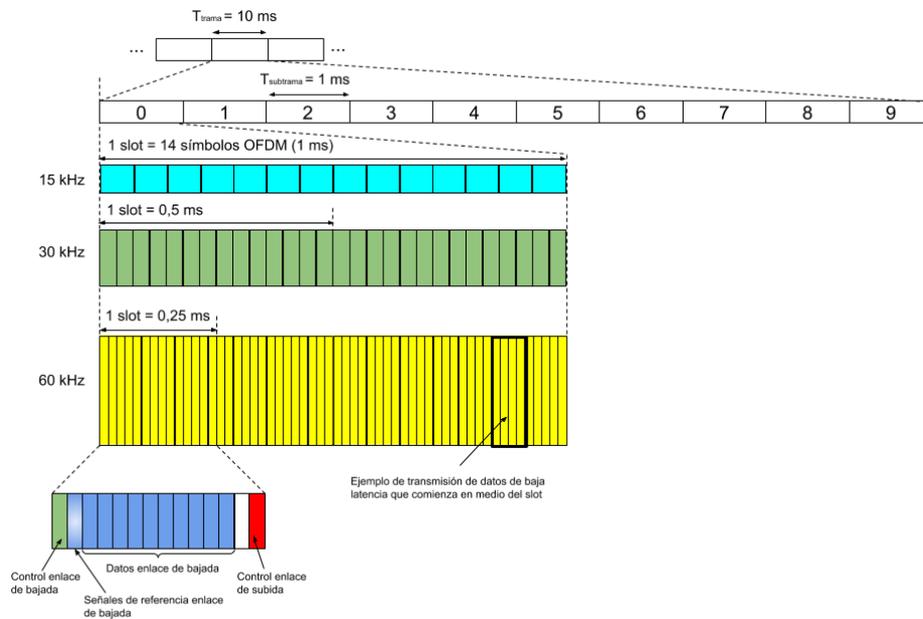


Figura 2.5: Estructura de la trama de 5G para diferentes numerologías y ejemplo de minislot [4].

posible comenzar una transmisión sin tener que esperar al comienzo del siguiente slot. El minislot resulta ser la mínima resolución que presenta la capa MAC del scheduler en el dominio del tiempo y para bandas de frecuencia mayores a 6 GHz el minislot soporta un largo de 1 símbolo OFDM [30]. En general los minislots pueden tener un largo desde 2 a 13 (largo del slot menos uno) símbolos OFDM y para algunas implementaciones de servicios de baja latencia (URLLC) los largos de minislot soportados son 2,4 y 7 símbolos [29]. NR soporta scheduling dinámico con múltiples TTI, lo que significa que es posible tener resolución temporal de minislot, de slot o de conjunto de slots.

2.2.4. HARQ, esquema de duplexación y esquemas de modulación

De manera de corregir errores se implementa una retransmisión del tipo HARQ (Hybrid Automatic Repeat-Request) donde el dispositivo reporta a la radio base sobre la decodificación de la transmisión recibida. En caso de que exista algún error se retransmiten los datos [4].

El esquema de duplexación se divide en dos partes: para las bandas de frecuencias bajas se suele implementar en tecnologías similares FDD (Frequency Division Duplex) mientras que para las bandas de frecuencias altas se utiliza TDD (Time Division Duplex). Para los casos de FDD se utiliza espectro apareado [4] lo que significa que en la banda de frecuencias correspondientes al enlace de bajada todos los slots son para downlink y en la banda correspondiente al enlace de subida todos los slots son para uplink. Es importante que se soporte FDD para lograr compatibilidad con las bandas bajas, donde no se utiliza TDD. Las bandas bajas son para

2.2. Telefonía Móvil de Quinta Generación NR (5G)

cobertura con mayores distancias entre BS donde TDD tiene algunos puntos en contra (por ejemplo el requerimiento de delay en la onda electromagnética). Para el caso de TDD se implementa espectro no apareado (los slots pueden ser bidireccionales comenzando con transmisiones de bajada seguida por transmisiones de subida [31]).

Los esquemas de modulación en NR son similares a LTE incluyendo BPSK (Binary Phase Shift Keying), QPSK (Quadrature Phase Shift Keying) y 16-QAM (16-Quadrature Amplitude Modulation), 64-QAM (64-Quadrature Amplitude Modulation) y 256-QAM (este último no está disponible en las primeras versiones de LTE).

2.2.5. Canales físicos y señales de sincronización

SS (Synchronizaton Signals) y PBCH (Physical Broadcast Channel)

El SS permite la sincronización tanto en el dominio de la frecuencia como en el dominio del tiempo y adquiere la sincronización con el PBCH. También le brinda al usuario información sobre la celda a la que pertenece a través del PCI (Physical layer Cell ID). El SS está dividido en primario (PSS) y secundario (SSS), entre ambos pueden identificar un total de 1008 celdas físicas. La combinación de SS y PBCH se conoce como SSB, cada SSB es mapeado con 4 símbolos en el dominio del tiempo y 20 RBs en frecuencia [32].

PRACH (Physical Random Access Channel)

El canal PRACH es utilizado para transmitir el preámbulo creado de forma aleatoria para permitir el acceso del usuario a la radio base y así sincronizarse. El preámbulo soporta dos secuencias de largos diferentes. La secuencia de largo 839 soporta 4 formatos de preámbulo (análogo a LTE) orientados a escenarios con celdas grandes. Estos preámbulos pueden ser usados en el RF1 y tienen un espaciado entre subportadoras de 1,25 o 5 kHz. Para la secuencia corta, de largo 139, existen 9 preámbulos diferentes, que apuntan principalmente a escenarios con celdas normales y pequeñas (ambientes interiores). Los preámbulos con formato corto pueden ser usados tanto en el RF1 con espaciado entre subportadoras de 15 o 30 kHz como en el RF2 con espaciado entre subportadoras de 60 o 120 kHz [32].

Canales físicos compartidos

El contenido del PDSCH (Physical Downlink Shared Channel) son los transport blocks que corresponden a datos del usuario, paging, broadcast y señalización del usuario en el sentido de bajada. Cada transport block contiene un CRC (Cyclic Redundance Check) para detectar errores, seguido de una fase LDPC^v (Low Density Parity Check) que se aplica sobre cada bloque de código y a partir de estos

^vPara soportar tasas muy altas en el enlace de bajada de datos se codifica el canal mediante códigos LDPC.

Capítulo 2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

bloques de código se crea un codeword a transmitir por el PDSCH. El PDSCH puede contar con dos codewords para soportar una transmisión de hasta 8 layers [32]. El PUSCH (Physical Uplink Shared Channel) se utiliza para transmitir los datos de información de subida.

PDCCH (Physical Downlink Control Channel)

El scheduler reside en la BS (Base Station) o gNB (gNodeB) y toma decisiones basado en reportes de la calidad del canal obtenido de los dispositivos. También se priorizan diferentes tráficos según los requerimientos de calidad de servicio que presente cada dispositivo. El conjunto de información de control del scheduling se denomina DCI (Downlink Control Information) y es transportado en el PDCCH. Los símbolos que conforman el PDCCH son mapeados a recursos físicos en unidades denominadas CCEs (Control Channel Elements). Cada CCE consiste en 6 REGs (Resource Element Groups), donde un REG es definido como un PRB en un símbolo OFDM que contiene nueve REs del payload del PDCCH y tres DMRS (Demodulation Reference Signal) REs. En cada DCI se pueden agrupar 1,2,4,8 o 16 CCEs, y el número de CCEs por DCI se define como AL (Aggregation Level) [33]. Para tráficos que requieran baja latencia es posible configurar más de un PDCCHs por slot, aunque lo normal es que haya uno por slot.

Un CORESET (Control Resource Set) contiene los parámetros necesarios (recursos en el dominio de la frecuencia y el tiempo y secuencia de scrambling para los DMRS) para mapear los DCI con determinado AL en un BWP (Bandwidth Part: número de PRBs consecutivos asociados a una subportadora) dado. Cada usuario puede estar configurado con hasta 3 CORESETs por BWP y puede tener asignados hasta 4 BWP, por lo que cada usuario puede tener configurados hasta 12 CORESETs que se enumeran de 0 a 11 [33]. El CORESET con índice 0 (CORESET 0) tiene una configuración especial [33] y diferente a los demás los cuales presentan 6 PRBs en frecuencia y uno, dos o tres símbolos OFDM en el dominio del tiempo.

Señales de referencia

Las señales de referencia que utiliza la capa física son DMRS (downlink and uplink Demodulation Reference Signals) y PTRS (downlink and uplink Phase-Tracking Reference Signals), CSI-RS (Channel-State Information Reference Signals) y SRS (Sounding Reference Signals) [32].

2.2.6. Convivencia entre servicios eMBB y URLLC

Teniendo en cuenta los servicios que atiende la red 5G, la mayor parte del tráfico será correspondiente a usuarios eMBB y servicios mMTC (este último no es tratado en el presente documento, se hace foco en los servicios eMBB). El tráfico URLLC es intermitente: se caracteriza por producirse en ráfagas y le insume a la red una pequeña porción de la carga del tráfico total. Otra particularidad del tráfico URLLC es que debe ser atendido con prioridad para respetar los requerimientos de baja latencia. Estos servicios, al pretender baja latencia en las comunicaciones

2.2. Telefonía Móvil de Quinta Generación NR (5G)

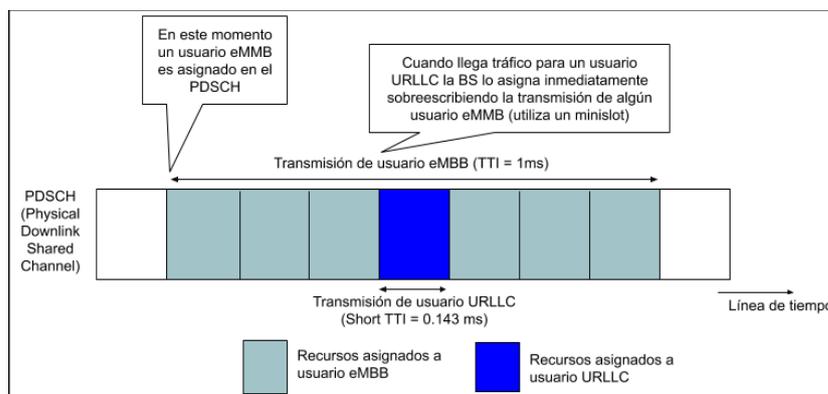


Figura 2.6: Funcionamiento del punctured scheduler.

punta a punta deben ser robustos con un BLER (BLock Error Ratio)^{VI} menor al de los usuarios eMBB, por lo que la eficiencia espectral de las modulaciones para este tipo de usuarios será inferior a la de un usuario eMBB.

La manera más sencilla de atender los servicios URLLC es reservando recursos exclusivos para los usuarios de este tipo lo que produce una sub utilización de los recursos ya que los mismos están vacíos la mayor parte del tiempo.

Buscando un adecuado rendimiento de la red, 5G introduce varias mejoras para disminuir latencia y en particular para compartir espectro con usuarios eMBB y URLLC. Entre los que se quiere destacar el Puncturing y el Network Slicing.

Punctured Scheduling

Para estos casos es que 5G cuenta con la opción de Punctured Scheduling (asignación de recursos “pinchada”) que se basa en sobrescribir parte de la transmisión de algún usuario eMBB [7] [8]. Para poder implementar Punctured Scheduling el TTI para el tipo de usuarios URLLC se utilizarán los sTTI (short TTI) o minislot que equivale a por lo menos 2 símbolos OFDM. A modo de ejemplo: para un espaciado entre subportadoras de 15 kHz el tráfico de los usuarios eMBB se organiza en slots de 1 ms (abarca 14 símbolos OFDM) mientras que el tráfico correspondiente a los usuarios URLLC se puede arreglar en minislots de 0.143ms [7] [8]. El ejemplo se ilustra en la Figura 2.6.

Basado en la idea de que la radio base atiende a usuarios eMBB en el enlace de bajada con el tiempo dividido en TS (Time Slots), durante la transmisión puede suceder que un usuario con servicio del tipo URLLC requiera recursos. Si el usuario fuera tratado de forma normal debería esperar a que finalice el actual time slot para que se le pueda asignar un recurso, lo cual perjudicaría el servicio brindado y sería difícil cumplir con los requerimientos de latencia estrictos. Los usuarios URLLC asignados mediante puncturing tienen el servicio de baja laten-

^{VI}El BLER es la tasa de error de un RB, no confundir con la tasa de error de bit, o de símbolo. En un RB tenemos 168 símbolos para un TTI (o 24 símbolos para un sTTI) donde la cantidad de bits que pueden tener dichos símbolos es variable.

Capítulo 2. Red 5G NR (New Radio) y transición con LTE (Long Term Evolution)

cia y baja probabilidad de error, a expensas de perjudicar el rendimiento de los usuarios eMBB a los cuales se le quitan los recursos y del espacio ocupado por la señalización en el PDCCH requerida para señalar dicha asignación, la implementación de esta señalización queda a cargo de los fabricantes ya que no se encuentra estandarizada al momento. Para notificar al usuario eMBB frente a una transmisión con puncturing se envía un mensaje HARQ para que la radio base informe al usuario que se debe retransmitir parte del mensaje.

Network Slicing

El Network Slicing busca asegurar la personalización de los servicios apuntando al aislamiento y separación lógica y física de los recursos de la red pertenecientes a una infraestructura física común. La posibilidad de crear (bajo demanda y de manera programable) segmentos de red de extremo a extremo rentables y dedicarlos al aprovisionamiento dinámico de diversos servicios se considera una característica importante en 5G [5]. Las slices se componen de VNFs (Virtual Network Functions) tienen características particulares según cual sea el servicio al cual estén asociadas (comunicaciones en tiempo real, servicios de baja latencia, entre otros).

Para utilizar Network Slicing como solución al problema de la convivencia entre servicios eMBB y URLLC se necesita que una misma BS pueda manejar zonas del espectro con distintas numerologías. Además se deben resolver dos problemas relacionados a la asignación de recursos: la asignación de recursos por slice y la asignación de usuarios por slice (un método posible para realizar la asignación es presentado por Yu Abiko en su paper [34]).

En el presente documento se aborda el problema de la convivencia entre servicios eMBB y URLLC desde la perspectiva del Punctured Scheduling (quedando por fuera del alcance del proyecto el otro abordaje).

2.2.7. Beamforming

Para ampliar la cobertura en altas frecuencias se utiliza beamforming, lo cual consiste en un arreglo de antenas de una cantidad elevada de elementos, donde se introduce un defasaje en cada uno tal que se pueda direccionar el haz resultante en la dirección deseada (ver Figura 2.7). La zona de cobertura del sector se divide en varios haces (beams) y cada UE (User Equipment) es servido a través de un haz específico. Para lograr esto es necesario realizar un barrido por todos los haces y repetir la misma señal en múltiples símbolos OFDM. Es posible sintetizar un haz particular para cada usuario y no de entre un set de haces predefinidos entre los cuales se dividió el sector.

La información necesaria para el uso de estos esquemas de múltiples antenas se obtiene de reportes basados en la transmisión de señales de referencia y en información del estado del canal CSI (Channel State Information) para el enlace de bajada. Mientras que para el enlace de subida se utilizan solo las señales de referencia sin CSI.

2.3. Resumen del Capítulo

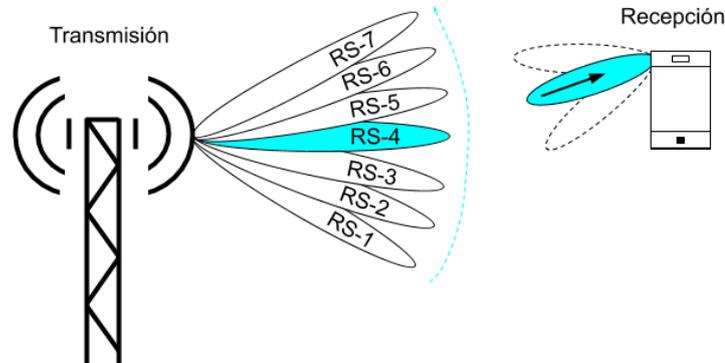


Figura 2.7: Beamforming: el dispositivo puede medir en un conjunto de RS (Reference Signals) correspondientes a diferentes haces de enlace descendente con el objetivo de refinar la transmisión [35].

2.2.8. MIMO masivo

En las bandas de frecuencias bajas se utiliza MIMO (Multiple-Input Multiple-Output) para evadir los problemas de interferencia causados por la separación entre antenas. Consiste en contar con un arreglo de antenas para recibir y otro arreglo de antenas para transmitir de manera de obtener un multiplexado espacial (se hace posible explotando la alta resolución del CSI del feedback del canal y utilizando vectores DFT - Discrete Fourier Transform-). En NR se especifican 12 señales de referencia para demodulación en transmisiones MIMO y cada dispositivo soporta hasta 8 capas MIMO en el enlace de bajada y hasta 4 en el enlace de subida [4]. Para un análisis más detallado sobre MIMO y beamforming se recomienda el capítulo 12 “Beam Management” [35] del libro de referencia “5G NR: the Next Generation Wireless Access Technology” [36].

2.3. Resumen del Capítulo

En el presente capítulo se expuso un breve resumen de la transición entre LTE y 5G. Además se explicaron las diferentes novedades que introduce la red de telefonía 5G (principalmente de las releases 15 y 16).

A continuación se desarrollará un capítulo sobre la teoría del scheduling (asignación de recursos) que sirvió de base teórica para el proyecto.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 3

Scheduling: Asignación de recursos

Dado que la mayor parte del proyecto está relacionada con lo que se llama scheduling o asignación de recursos, se debe aclarar y establecer términos sobre este concepto ya que aparece en varias ocasiones a lo largo del documento, sobre todo en la sección de las implementaciones.

La asignación de recursos dinámica, o “scheduling”, es un problema muy similar al planteado en economía con el mismo nombre, donde dichos recursos son escasos. En estos contextos aparecen los conceptos de “justicia” y “eficiencia”. Dado que los usuarios no son iguales, el reparto en cantidades iguales no necesariamente significa justicia. Es claro que el concepto de justicia es difícil de definir en términos generales, en teoría de juegos [37] se pueden encontrar algunas definiciones así como también de la función bienestar social (social welfare). En los algoritmos de asignación de recursos lo que se busca es maximizar dicha función. Análogamente se tiene la eficiencia, donde el objetivo es maximizar la producción de un bien, o un servicio. Estas definiciones son parte de los fundamentos que existen detrás de los algoritmos que pueden ser utilizados para regular el precio de un bien en el mercado, y en el contexto del proyecto serán aplicados para regular y administrar los recursos en una red de comunicaciones.

Este capítulo se remonta brevemente a los orígenes de este tema y sus primeros planteos en las redes de comunicaciones, seguido de criterios para medir la eficiencia y la justicia en las asignaciones, se realizarán algunas observaciones finalizando con algunos ejemplos de algoritmos y sus principales ventajas y desventajas.

3.1. Orígenes

En temas de asignación de recursos y para plantear el problema de optimización asociado, se define la función de utilidad $U_i(x_i)$. Esta busca representar la conformidad del usuario i (del conjunto de N usuarios) respecto a los recursos asignados x_i , que podrían ser cualquier cosa que sea de utilidad para los usuarios. En las redes de comunicaciones suele ser la tasa alcanzable. Esta es una función análoga a la función de bienestar social del área de economía.

Las primeras apariciones en publicaciones sobre asignación de recursos en redes

Capítulo 3. Scheduling: Asignación de recursos

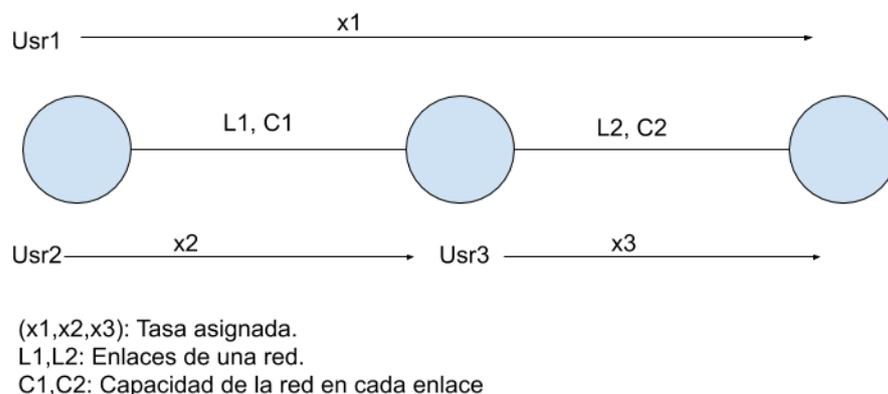


Figura 3.1: Conjunto de enlaces de capacidad limitada que es asignada a los usuarios.

fueron en la década de los noventa [38]. Más precisamente en 1998, Frank Kelly publicó lo que se considera el origen del estudio del problema de asignación de recursos en una red lineal de varios enlaces [39] como la de la Figura 3.1. Kelly plantea el problema de maximizar la utilidad de las tasas de los usuarios. Una hipótesis importante que se asume en el estudio del problema es que los usuarios poseen un tipo de tráfico denominado tráfico elástico. Este es un tipo de tráfico que reacciona frente a cambios en la red, pudiendo adaptar su servicio a asignaciones de tasas o recursos variable, dándole al usuario utilidad a pesar de que podría ser mayor. Este es el tráfico más usual que circula en las redes de computadoras y en general se lo asocia al servicio estándar de HTTP (Hypertext Transfer Protocol) sobre TCP (Transmission Control Protocol), como puede ser descargar el contenido de un sitio web. También existe lo que se denomina tráfico inelástico que (a diferencia del tráfico elástico) no reacciona frente a la red. Este es un tipo de tráfico más exigente y más difícil de garantizar su funcionamiento esperado cuando se tiene una QoS (Quality of Service) inferior a la requerida, como suelen ser los requerimientos relacionados al retardo y jitter. Por ejemplo, en aplicaciones de tiempo real como suelen ser las de multimedia sobre IP, asociadas más a RTP (Real-time Transfer Protocol) como VoIP (Voice over IP), aunque también hay servicios de HTTP con este perfil de tráfico como lo es VoD (Video on Demand) [40].

Puede verse gráficamente en la figura Figura 3.2 qué aspecto tiene cada tipo de tráfico y cómo se diferencian. Notar que la función de utilidad para el tráfico elástico es infinitamente derivable, estrictamente creciente y cóncava, las cuales son propiedades de utilidad en problemas de optimización.

Como se verá el problema que plantea Kelly lo resuelve, da soluciones y estudia su estabilidad. El problema que se aborda en el proyecto, al tener un sólo enlace a considerar a priori parece ser más simple, sin embargo más adelante se verá que son necesarias otras simplificaciones para lograr un problema tratable.

Volviendo a Kelly, él plantea el siguiente problema de optimización [39]:

3.2. Criterios de justicia

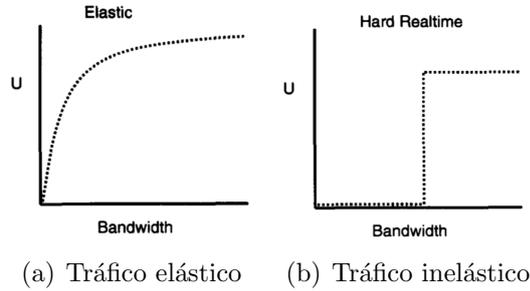


Figura 3.2: Utilidad para tráfico elástico y tráfico inelástico en función del ancho de banda. Figura extraída de [41].

$$\max_x \sum_{i=1}^N U_i(x_i) \quad (3.1a)$$

sujeto a:

$$\begin{aligned} x &\in C \\ x &\geq 0. \end{aligned} \quad (3.1b)$$

Donde C es la región de capacidad en la que vive x , estas restricciones son lineales y representan la capacidad de cada enlace. La región que definen sus ecuaciones es convexa. Por ejemplo, en el grafo de la Figura 3.1 las restricciones serían $x_1 + x_2 \leq C1$ y $x_1 + x_3 \leq C2$, y se puede ver que es una región convexa. De aquí es fácil ver que agregarle enlaces y usuarios no hace que las restricciones dejen de ser una región convexa. Este problema tiene solución factible siempre y cuando la región definida por x sea convexa y las funciones de utilidad sean estrictamente cóncavas. Se resuelve aplicando una relajación de Lagrange. Las hipótesis sobre la función de utilidad (cóncava y estrictamente creciente) y la restricción (función convexa) garantizan un máximo global. Esto significa que la solución es factible y es única. Más adelante se retomará este planteo y se realizarán algunas observaciones y resultados.

3.2. Criterios de justicia

Las definiciones de justicia provienen de la teoría de juegos mencionada previamente. Hay diferentes criterios o requerimientos que debe cumplir una asignación para ser considerada justa, o medir justicia relativa a otra asignación distinta. Hay conceptos de teoría de juegos que ayudan a entender estos criterios [37]. A continuación se presentan algunos criterios de justicia útiles con ejemplos.

La optimalidad de Pareto es un criterio en el cual dada una asignación de recursos x , no se puede incrementar la asignación a un usuario sin decrementar los recursos a otro. En otras palabras todos los recursos disponibles deben estar

Capítulo 3. Scheduling: Asignación de recursos

asignados. Algunas asignaciones $x = (x_1, x_2, x_3)$ Pareto óptimas para el problema de la Figura 3.1 con $C_1=1$ y $C_2=2$, pueden ser: $x = (0, 1, 2)$, $x = (0, 25; 0, 75; 1, 75)$ o $x = (0, 5; 0, 5; 1, 5)$, donde es fácil ver que para aumentar la tasa de algún usuario necesariamente se debe bajar la de otro porque los enlaces están saturados. También se puede ver que si bien las tres asignaciones son Pareto óptimas, no parecen ser igualmente justas. Por esta razón es necesario considerar otros criterios de justicia.

El criterio de justicia más básico es el de max-min fairness [42]: un vector de asignaciones $x = (x_i : i = 1, \dots, N)$ factible ($x \in C$ y $x \geq 0$) es max-min fair si hay al menos dos usuarios i e i^* ($i, i^* \in 1, \dots, N$) con

$$x_i > x_{i^*} \quad (3.2)$$

tal que para cualquier otra asignación donde aumenta x_i dentro de sus límites, entonces disminuye x_{i^*} ($x_i \uparrow \Rightarrow x_{i^*} \downarrow$). Las características de la región de factibilidad garantizan la existencia y unicidad de este vector.

Una forma de obtener una asignación max-min fair es subiendo las tasas de forma equitativa hasta que uno de ellos llegue a su máximo, aplicando lo mismo para el resto y así sucesivamente hasta alcanzar la capacidad máxima (water filling). De esta manera la única forma de seguir aumentando las tasas, es quitándosela a otro que tiene menos, debido a que ya alcanzaron su máximo. Si se aplica este algoritmo al problema de la Figura 3.1 se obtiene que $x_1 = 0,5$, $x_2 = 0,5$ y $x_3 = 1,5$. Este es un criterio que prioriza a los usuarios con tasas menores sin importar la diferencia entre los mismos, dado que si se cumple la ecuación (3.2), ningún incremento en x_i podrá compensar el decremento en x_{i^*} . Esto significa que este criterio no considera la proporción entre cuánto se beneficia el usuario i y cuánto se perjudica i^* .

Otro criterio que favorece a los usuarios en inferioridad de condiciones mencionado por Kelly en sus publicaciones [39] [38] es el llamado proportionally fair (proporcionalmente justo en inglés). Un vector de asignaciones factible $x = (x_i : i = 1, \dots, N)$, es proportionally fair si para cualquier otra asignación factible x^* , se cumple que:

$$\sum_{i=1}^N \frac{x_i^* - x_i}{x_i} \leq 0. \quad (3.3)$$

En otras palabras, una asignación es proportionally fair si se cumple que para cualquier otra asignación, los cambios proporcionales respecto a la asignación original son menores o iguales a cero.

Los criterios mencionados anteriormente especifican en contextos muy analíticos, lo cual es de gran utilidad para demostrar resultados y propiedades que debe cumplir una cierta asignación, no son muy útiles en contextos numéricos ya que en la práctica podría ser de interés comparar la justicia de dos algoritmos distintos teniendo algún tipo de referencia. Por esta razón a la hora de medir la justicia no siempre es útil tener criterios que separen los algoritmos en dos grupos (justos e injustos). Existen diferentes formas de medir cuantitativamente la justicia. Una de ellas y la que se utilizará a lo largo de este documento es el índice de Jain [43].

3.2. Criterios de justicia

El índice de Jain fue creado para medir cuantitativamente la justicia en una asignación. Para conseguir dicho objetivo se plantearon ciertas propiedades que el índice debía cumplir, a diferencia de otros criterios.

- Independiente al tamaño de la muestra: El criterio debe poder aplicarse a cualquier cantidad de usuarios (mayor o igual a dos).
- Independiente a la escala: El índice no puede variar por una variación de escala en las muestras. En otras palabras, el índice no puede ser distinto en dos asignaciones que tienen las mismas proporciones, no importa cual sea la cantidad absoluta de recursos disponibles.
- Conjunto de posibles valores acotado: El índice debe estar comprendido entre 0 y 1. Donde 1 sería una asignación totalmente justa y 0 una asignación totalmente injusta. De esta forma se puede medir qué tan justa es una asignación.
- Continuidad: Cualquier cambio en la asignación debería verse reflejado en el índice. Por ejemplo, si para tres usuarios se tiene una asignación (1, 3, 5), debería resultar en un distinto índice que la asignación (1, 4, 5)

Estas propiedades son analizadas y comparadas respecto a distintos criterios populares en la literatura, concluyendo que los mismos no logran cumplir todas las propiedades planteadas simultáneamente. Con esta motivación, Jain llegó a la expresión que llamó el Fairness Index, el cual puede verse fácilmente que cumple con las cuatro propiedades mencionadas previamente. El mismo se calcula como

$$f(x) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (3.4)$$

donde los x_i son los recursos asignados al usuario i .

Puede ser interpretado de forma porcentual, por ejemplo un índice de 0,1 para una asignación se puede decir que para el 90 % de los usuarios, dicha asignación es injusta.

Considere el siguiente ejemplo donde se reciben doscientos pesos para distribuir entre cien personas. Se proponen dos asignaciones:

Asignación A

Darle dos pesos a cada persona. En este caso $x_i = 2$, $i = 1, 2, \dots, 100$.

$$f_A(x) = \frac{(\sum_{i=1}^{100} x_i)^2}{100 \sum_{i=1}^{100} x_i^2} = \frac{(100x_i)^2}{100^2 x_i^2} = 1.$$

La asignación A es totalmente justa.

Capítulo 3. Scheduling: Asignación de recursos

Asignación B

Ordenar a las personas mediante cierto aspecto (altura, edad, peso, etc...), darle veinte pesos a los primeros diez y el resto no recibe dinero. Por lo tanto

$$x_i = \begin{cases} 20 & \text{si } i = 1, \dots, 10 \\ 0 & \text{si } i = 11, \dots, 100 \end{cases}$$

y el índice queda

$$f_B(x) = \frac{(\sum_{i=1}^{10} x_i)^2}{100 \sum_{i=1}^{10} x_i^2} = \frac{(10x_i)^2}{100 * 10 * x_i^2} = 0,1.$$

Entonces, la asignación B es 10 % justa.

Este ejemplo muestra lo sencillo que es calcular el índice partiendo de una asignación y la poca información que este requiere.

3.3. Observaciones

Una observación importante es que para la función de utilidad $U_i(x_i) = \log(x_i)$, la solución factible x^* que resuelve el problema (3.1a) es el máximo global (dado por la convexidad de las restricciones y la concavidad de la función de utilidad) y además es proportionally fair [42].

La prueba de que es proportionally fair consiste en tomar la asignación óptima con las tasas x_i^* y otra asignación $x'_i = x_i^* + \delta_i$ donde $x_i^* \ll \delta_i$. Evaluándolo en el desarrollo de Taylor de primer orden en x_i^* :

$$\sum_{i=1}^N \log(x'_i) = \sum_{i=1}^N \log(x_i^*) + \sum_{i=1}^N \frac{1}{x_i^*} (x_i - x_i^*) + o(|x_i - x_i^*|^2) \quad (3.5a)$$

$x_i \rightarrow x'_i$

$$\Rightarrow \sum_{i=1}^N \log(x'_i) \approx \sum_{i=1}^N \log(x_i^*) + \sum_{i=1}^N \frac{\delta_i}{x_i^*} \quad (3.5b)$$

$$\Rightarrow \sum_{i=1}^N \frac{\delta_i}{x_i^*} \leq 0 \Leftrightarrow \sum_{i=1}^N \frac{x'_i - x_i^*}{x_i^*} \leq 0, \quad (3.5c)$$

$(\delta_i = x'_i - x_i^*)$

La ecuación (3.5b) se obtuvo de reemplazar x_i por x'_i en la ecuación (3.5a) y conservar hasta el término lineal. Observar luego que en la ecuación (3.5b), el término independiente de la aproximación es la función objetivo evaluada en el punto que la maximiza, por lo tanto, el término de primer orden necesariamente debe ser menor o igual a cero por la definición de máximo global. Finalmente, escribiendo δ_i en función de x_i^* y x'_i tal como se ve en la ecuación (3.5c), se llega a que la asignación que maximiza la función objetivo (x_i^*) es proportionally fair.

3.3. Observaciones

En el ejemplo de la Figura 3.1:

$$\max_x \sum_{r \in R} \log(x_r) \quad (3.6a)$$

sujeto a:

$$\begin{aligned} x_1 + x_2 &\leq C1 \\ x_1 + x_3 &\leq C2 \\ x &> 0. \end{aligned} \quad (3.6b)$$

Resolviendo para $C1 = 1$ y $C2 = 2$ se obtiene que $x_1 = \frac{\sqrt{3}+1}{3+2\sqrt{3}}$, $x_2 = \frac{1}{\sqrt{3}}$ y $x_3 = \frac{\sqrt{3}+1}{\sqrt{3}}$. Se puede ver que utilizando cualquier otra asignación x de las halladas en la sección anterior se cumple la ecuación (3.3) para $x_i^* = (\frac{\sqrt{3}+1}{3+2\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{\sqrt{3}+1}{\sqrt{3}})$.

Si en lugar de querer asignar las tasas de manera justa se quisieran asignar de manera eficiente utilizando toda la capacidad de cada enlace, las distribuciones resultantes serían injustas dado que algunos usuarios podrían quedar sin recursos asignados. En particular, si algún usuario utiliza más de un enlace mientras que los otros no, la única forma de maximizar la tasa del sistema es sin asignarle recursos a estos últimos usuarios. Para el problema de la Figura 3.1, la asignación más eficiente es $x_1 = 0$, $x_2 = 1$ y $x_3 = 2$. Se puede observar como el usuario que utiliza ambos enlaces no recibe nada.

Otra observación es que los criterios mencionados anteriormente usan la tasa como medida de justicia al asignar pero no consideran que las colas de paquetes de los usuarios o la demora que experimentan pueden afectar su utilidad. Existen algoritmos que dan el canal al usuario con la mayor relación Q_i/R_i , siendo Q_i la cola de datos del usuario i y R_i su tasa alcanzable. De esta manera se le da prioridad a los usuarios que tienen más datos pendientes.

Los algoritmos mencionados hasta ahora no consideran enlaces multi canal, que es el caso de OFDMA. Se puede escribir el problema de optimización para enlaces multi canal [44] con el que se maximiza la utilidad conjunta de los usuarios y se obtiene una asignación proportionally fair. Si se define la tasa del usuario i como:

$$x_i = b_i W \log\left(1 + \frac{P_i h_i}{b_i W N_0}\right) \quad (3.7)$$

suponiendo que la potencia P_i se reparte de forma equitativa para cada usuario, W es el ancho de banda de cada subcanal, b_i es la cantidad de subcanales asignados al usuario i y h_i es la respuesta del canal, medida y asumida que se suponen idénticas para todos los usuarios y subcanales con un ruido de densidad constante N_0 . Entonces el problema de optimización queda:

$$\max_x \sum_{i=1}^N U_i(x_i(t)) \quad (3.8a)$$

sujeto a:

Capítulo 3. Scheduling: Asignación de recursos

$$\begin{aligned}x &\in C \\ \sum_i b_i &\leq B \\ x &\geq 0.\end{aligned}\tag{3.8b}$$

Donde B es la cantidad de subcanales del sistema.

Dado que los b_i son coeficientes enteros, resolver este problema para $b_i \in \mathbb{Z}$ implica probar con todos los posibles b_i para todos los usuarios. Se puede relajar el problema para que los b_i sean reales de modo de resolver una aproximación y de esta forma queda un problema de optimización con restricciones convexas para lo cual ya se sabe que con estas hipótesis es un problema resoluble. Sin embargo, es un problema que hay que resolver en cada TTI, esto es, cada menos de un milisegundo. Hay muchas simplificaciones en este planteo. Por ejemplo, se asume que la atenuación no es selectiva en frecuencias, lo cual es una aproximación que no siempre es válida y depende mucho del factor W . La potencia se asume repartida por igual, repartir la potencia eficientemente también optimiza el rendimiento. El problema de optimización propuesto busca una asignación *proportionally fair* en cuanto a la tasa alcanzable y no en cuanto al throughput medio. El problema de optimización 3.8 es difícilmente resoluble en menos de un milisegundo para $b_i \in \mathbb{Z}$ y sin considerar las simplificaciones anteriores. Se ha investigado y resuelto el problema multi canal mediante métodos computacionales muy eficientes, precisando alrededor de 25 iteraciones para converger a la asignación óptima con una complejidad de orden lineal respecto a la cantidad de usuarios ($\mathcal{O}(n)$) [44].

A esta altura ya se puede asegurar que, los algoritmos planteados anteriormente proveen soluciones analíticas, que se tienen resultados que garantizan la existencia de dichas soluciones y que cumplen con los criterios de justicia establecidos. Sin embargo, estos algoritmos son difícilmente aplicables a una red inalámbrica como la red 5G, principalmente por su alto costo computacional. Por esta razón, actualmente en la práctica se utilizan otros algoritmos que, a diferencia de los mencionados anteriormente, calculan un índice o métrica (generalmente por cada subcanal) y se le asigna el subcanal al usuario que tenga el mayor índice. Este es el tipo de algoritmos sobre los que se trabaja en este proyecto.

3.4. Algunos algoritmos

En esta sección se mencionan y explican brevemente [45] algunos de los algoritmos más usados en la actualidad.

Round Robin

El algoritmo más sencillo y más usado en la actualidad es el RR (Round Robin), el cual consiste en un sistema FIFO¹ (First In First Out) cíclico. Esto implica

¹Se le llama FIFO a los sistemas en los que se ingresa un arreglo de elementos y éstos son devueltos por el sistema en el mismo orden que se ingresaron.

3.4. Algunos algoritmos

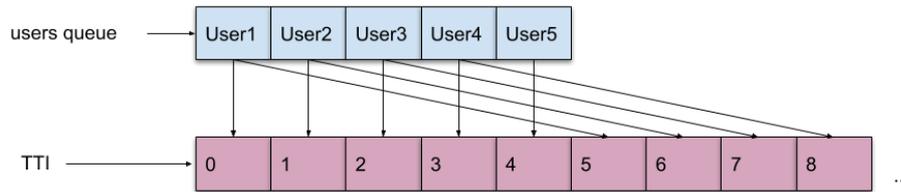


Figura 3.3: Esquema del algoritmo Round Robin para un solo canal.

que dicho algoritmo atiende a la cola de usuarios en el orden en el que llegan, recorriendo toda la cola hasta llegar al último, asignándole un recurso a cada usuario de la cola, y cuando llega al último se recorre la cola nuevamente hasta llenar la grilla para el TTI en el que se encuentra. Un esquema gráfico de este algoritmo está representado en la Figura 3.3 donde se le da el canal durante un intervalo de transmisión al primer usuario de la cola, luego al segundo y así sucesivamente hasta llegar al último y luego se repite la secuencia. Para el caso de tener varios subcanales, cada TTI tendrá más de un recurso disponible para asignar, formando así una matriz cuyas columnas corresponden a los intervalos de transmisión y las filas corresponden a los subcanales posibles.

Matemáticamente, se determina el índice del usuario en la cola que se le asignarán los recursos en el siguiente slot, $i[t + 1]$ como

$$i[t + 1] = i[t] + 1. \quad (3.9)$$

Recorriendo los índices de manera cíclica, se puede ver en la ecuación (3.9) cómo el usuario al que se le asignará el canal en el siguiente intervalo de transmisión será el que le sigue al que fue atendido en el instante actual. Entonces la manera de asignar es como la de la Figura 3.4, donde se va asignando mediante la misma lógica FIFO pero ocupando todos subcanales disponibles primero, y luego se continúa asignando en el siguiente intervalo de tiempo, retomando el estado en el que quedó la cola al finalizar la asignación en el intervalo anterior. En el caso de múltiples subcanales se puede ver con más facilidad como se va actualizando la cola cada vez que se asigna un recurso.

Se puede concluir finalmente que este algoritmo no busca maximizar la utilidad de ningún parámetro de los usuarios, sino que va asignando de a un recurso por usuario sin importarle la utilidad que tiene la tasa (que pueda alcanzar con dicho recurso). Por esta razón no es considerado un algoritmo “justo” en el sentido de que la tasa no se reparte de forma equitativa necesariamente, ya que las tasas alcanzables de los usuarios no tienen por que ser iguales para una misma unidad de recurso asignable. Entonces, la distribución equitativa de resource blocks no implica una distribución equitativa de la tasa. En cuanto a la eficiencia, este algoritmo puede resultar en un bajo throughput, dado que la política es un recurso por usuario sin importar su capacidad.

Capítulo 3. Scheduling: Asignación de recursos

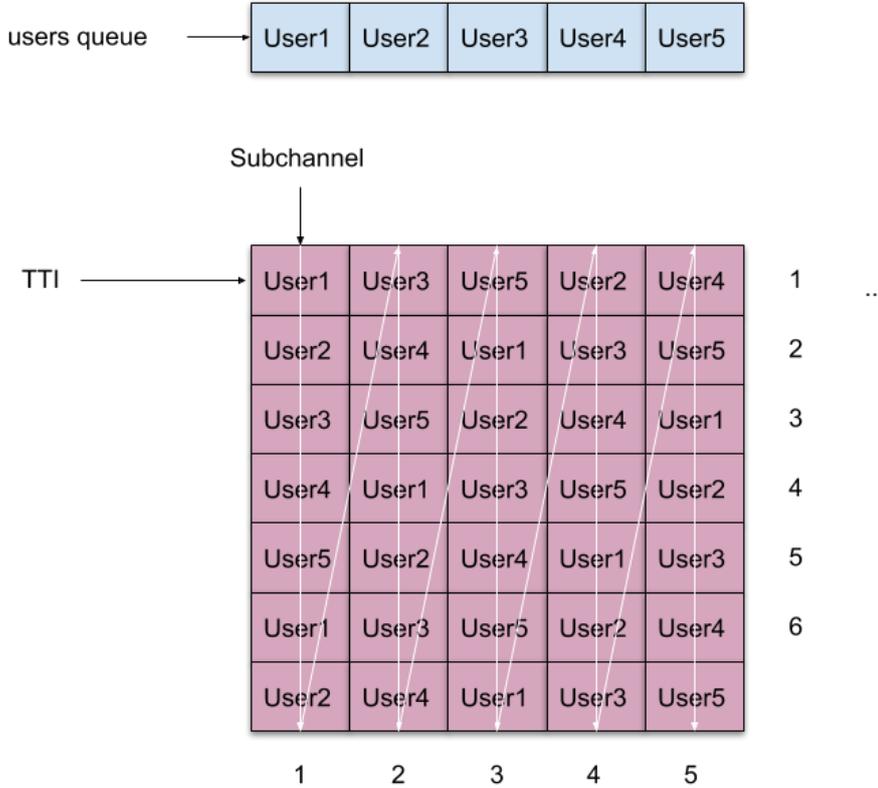


Figura 3.4: Esquema del algoritmo Round Robin para siete subcanales.

Max Throughput

Otro algoritmo muy usado para la asignación de recursos es el de máximo throughput, en el simulador Vienna 5G SLS se puede encontrar un algoritmo de este tipo con el nombre de BestCQI (Best Channel Quality Indicator).

Como lo dice su nombre, este algoritmo busca maximizar throughput del enlace entonces se le debe asignar el canal al usuario que tenga la mayor tasa alcanzable, obteniendo de esta manera un mayor throughput en la red.

Si se le asigna el canal al usuario i , el throughput instantáneo esperado para este slot sería

$$\hat{r}_i[t] = \frac{\hat{\eta}_i[t]}{T_s}, \quad (3.10)$$

donde T_s es la duración del time slot y $\hat{\eta}_i[t]$ es la cantidad estimada de bits que se pueden transmitir libre de errores. El throughput total esperado en la red sería

$$\hat{r}[t] = \sum_{i=0}^{M-1} \hat{r}_i[t] I(i), \quad (3.11)$$

donde $I(i)$ es un indicador el cual vale 1 si al usuario i se le asignaron recursos y

3.4. Algunos algoritmos

0 si no. El objetivo es asignarle el canal al usuario que maximice el throughput total de la red, entonces se debería asignar el canal al usuario con el throughput esperado más alto para maximizar $\hat{r}[t]$.

Una forma de estimar $\hat{r}_i[t]$ es [45]

$$\hat{r}_i[t] = W \log_2 \left(1 + \frac{\Gamma_i[t]}{\theta} \right), \quad (3.12)$$

donde W es el ancho de banda, θ el factor que establece el gap entre la capacidad de canal y el MCS (Modulation Coding Scheme) disponible más próximo y $\Gamma_i[t]$ es el SINR (Signal to Interference plus Noise Ratio) en el tiempo t dada la potencia asignada. Claramente el usuario con el mayor throughput estimado $\hat{r}_i[t]$ es también el usuario con el SINR ($\Gamma_i[t]$) más alto. Por esta razón también se lo conoce como el algoritmo máximo SINR. Dado que las variaciones en el canal hacen variar la métrica de los usuarios, se dice que el algoritmo de máximo throughput es un algoritmo que depende del estado del canal.

El CQI es un índice que cumple dicha función, siendo equivalente al máximo MCS posible dadas las condiciones del canal y la tasa de error máxima que se puede soportar. Por lo tanto mientras mayor sea el CQI de un usuario, mayor será su tasa alcanzable. Ya se vio en la sección anterior que este tipo de algoritmos llevan a resultados injustos.

Proportional Fair

Finalmente se presenta el algoritmo llamado PF (Proportional Fair), el cual es un algoritmo que busca maximizar el throughput de la red manteniendo una política de proveer a todos los usuarios de una utilidad mínima entonces ningún usuario se queda sin usar el canal. El objetivo del Proportional Fair es maximizar la función de utilidad conjunta

$$\sum_{i=0}^{M-1} \log(S_i) \quad (3.13)$$

donde S_i es el throughput a largo plazo del usuario i , y la función de utilidad $U_i(S_i) = \log(S_i)$. S_i puede cambiar en cada time slot; se define el throughput en el slot $t - 1$ como $S_i[t - 1]$. $S_i[t]$ se puede predecir usando un suavizado exponencial¹¹.

$$\hat{S}_i[t] = \left(1 - \frac{1}{\tau}\right) S_i[t - 1] + \frac{1}{\tau} \hat{r}_i[t] I(i), \quad (3.14)$$

donde $\tau \gg 1$. Puede verse que para maximizar $\sum_{i=0}^{M-1} \log(S_i)$, se debe asignar el canal al usuario con la relación

$$\frac{\hat{r}_i[t]}{S_i[t - 1]} \quad (3.15)$$

¹¹El suavizado exponencial es un filtro discreto de primer orden, producto de discretizar la respuesta de un filtro pasabajos RC.

Su respuesta en el tiempo es $y_{k+1} = \alpha x_{k+1} + (1 - \alpha)y_k$, con $0 < \alpha < 1$.

Capítulo 3. Scheduling: Asignación de recursos

más alta.

La razón de su nombre es porque la asignación resultante cumple con el criterio de proportionally fair. Es fácil ver que esto se cumple, dado que la función de utilidad es el logaritmo y ya fue demostrado en la sección anterior que la asignación que maximiza la utilidad conjunta de los usuarios cuando la función de utilidad es el logaritmo es proportionally fair.

Existen muchas variantes de este algoritmo que dan mejores resultados para determinados escenarios, donde la métrica resulta en expresiones similares a (3.15).

3.5. Resumen del Capítulo

Este capítulo deja una idea de lo que es el scheduling y la razón por la que existe en las redes de datos y cómo influye en el desempeño de éstas. También deja una noción sobre justicia y sus criterios, eficiencia y el compromiso entre ellos. Algo importante que se debe recordar sobre este capítulo es que el problema de asignación de recursos como un problema de optimización puede llegar a ser muy complejo si no se toman simplificaciones y es computacionalmente costoso por lo que usualmente se buscan algoritmos en los que se calcula un índice por usuario y se le asigna el canal al que tenga el mayor de estos índices. Este tipo de algoritmos resuelven más rápido el problema, si bien no siempre optimizan el uso del canal o la utilidad conjunta de los usuarios.

Capítulo 4

Simulador Vienna 5G SLS

El software de simulación es una herramienta fundamental para el desarrollo e investigación de las redes inalámbricas. Existen distintos tipos de simuladores: LLS (Link Level Simulator), SLS (Sistem Level Simulator) y NLS (Network Level Simulator). Los LLS comprenden el estudio y simulación de las características del enlace entre un transmisor y un receptor. El principal objetivo de estos simuladores es medir la probabilidad de error introducida por el canal de acuerdo a modelos de capa física incluyendo modulación y codificación del canal [46]. Los SLS se encargan de simular escenarios con muchas BS y gran número de usuarios registrados en las mismas. Permiten estudiar lo que sucede en promedio con los usuarios de la celda, evaluando parámetros de la red como capacidad, latencia, cobertura, eficiencia [46]. Además estos simuladores permiten obtener información del comportamiento que existe en los bordes de la celda. Son útiles para la planificación de la red y también permiten el análisis de nuevos métodos o características como por ejemplo la asignación de recursos (esta será el punto de interés del presente documento). Los NLS se encargan de analizar elementos de la red y los protocolos correspondientes al núcleo y al acceso como por ejemplo el RLC (Radio Link Control), el PDCP (Packet Data Convergence Protocol) o el RRC (Radio Resource Control) [46]. Al implementar al detalle todas las capas del stack de protocolos, una simulación (en un simulador NLS) de varios usuarios y varias BS demora mucho tiempo y requiere gran costo computacional.

El simulador utilizado es el Vienna 5G SLS [13] (es un simulador del tipo SLS) desarrollado por la TU Wien (Universidad Tecnológica de Vienna) disponible desde Octubre del 2018. Al mismo se accede mediante una licencia académica otorgada por TU Wien. La principal finalidad del simulador es la investigación de redes en gran escala basándose en su geometría y describiendo los efectos de propagación mediante los modelos apropiados (funciona sólo para downlink). Hasta el momento existe una única versión (release 1.0) la cual hereda funcionalidades del lanzamiento previo Vienna LTE-A System Level Simulator [47] (simulador de la TU Wien para redes 4G LTE-A). El Vienna 5G SLS está implementado sobre Matlab (versión 2018 en adelante) y utiliza POO (Programación Orientada a Objetos). Desde mayo de 2020 se encuentra disponible el simulador Vienna 5G Link Level Simulator [48] (también disponible bajo licencia) el cual permitirá brindar insumos al 5G SLS para

Capítulo 4. Simulador Vienna 5G SLS

enriquecer su funcionamiento. Por ejemplo a partir del LLS se podrían ampliar las tablas de SINR-BLER (Signal to Noise Ratio - Block Error Rate) necesarias para utilizar MCS de 256QAM.

En el presente capítulo se describe el funcionamiento del simulador analizando las cuatro partes en las que se encuentra organizado (inicialización, pregeneración, bucle principal de simulación y post procesamiento). Además se explica el proceso necesario para realizar una simulación comentando las funciones principales involucradas.

4.1. Estructura del simulador

El simulador tiene como entradas un escenario y un conjunto de parámetros, en base a esto se genera un entorno y se simulan las transmisiones. El simulador se puede dividir en cuatro bloques según su funcionalidad: inicialización, pregeneración, bucle principal de simulación y post procesamiento. En la Figura 4.1 se puede observar un diagrama donde se muestran dichos bloques con las diferentes funciones que realiza el simulador en cada uno de ellos. A continuación se dará una descripción de cada uno de los bloques (basados en la sección 3: “Simulator Structure” del paper de referencia del simulador Vienna 5G SLS: “Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator” [13]).

4.1.1. Inicialización

Inicialización de parámetros

Antes de comenzar una simulación es necesario definir todos los parámetros a utilizar, para ello el simulador cuenta con el paquete “+parameters” donde se incluyen los diferentes parámetros agrupados por tipo. En los subpaquetes “+basestation” y “+user” se inicializan los parámetros de los elementos de la red. Por ejemplo dentro de “basestation” se puede elegir si las BS estarán dispuestas en

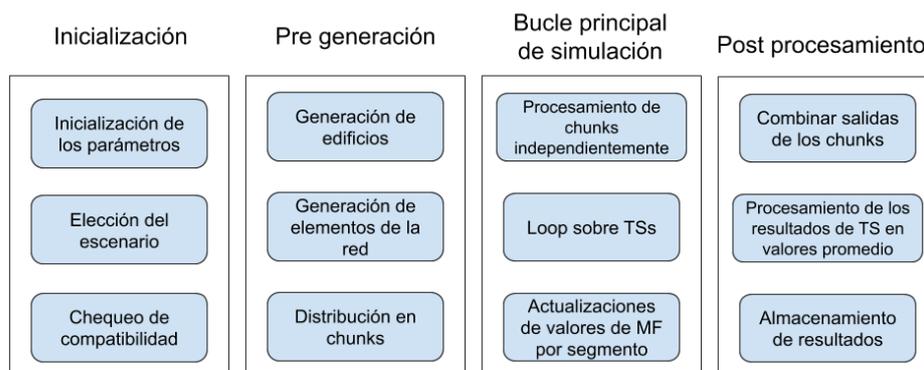


Figura 4.1: Bloques del simulador Vienna 5G SLS.

4.1. Estructura del simulador

cuadrícula o en anillo y dentro del paquete “+antennas” es posible elegir el tipo de antena a utilizar (omnidireccional, de tres sectores, de seis sectores). Además se puede elegir la cantidad de antenas receptoras y transmisoras por BS. Mientras que dentro de “+user” se puede elegir la cantidad de antenas receptoras y transmisoras por dispositivo, el tipo de movimiento de los usuarios y el tipo de modelo de tráfico que utilizará el usuario (implementación que se describe en el Capítulo 5). En los subpaquetes “+building”, “+city” y “wall” se definen los parámetros que describen los edificios (altura, largo, ancho y pérdidas que introduce por interferencia) creados para formar las ciudades para una simulación. En el subpaquete “+regionOfInterest” se define la región de la simulación en un plano tridimensional. Los parámetros relativos a la capa MAC y física se encuentran en los subpaquetes “+resourceGrid” y “+transmissionParameters” respectivamente. En “resourceGrid” originalmente se encontraba el archivo “LTE.m” que definía la grilla de recursos según LTE: con slots de 7 símbolos OFDM agrupados de a dos formando la subtrama de LTE [49] (lo que coincide con el slot de 5G en la numerología con espaciado de subportadora de 15 kHz) con 14 símbolos OFDM en una duración de 1 ms. En este paquete se agregaron los archivos: “NR_minislot.m” y “NR.m”. El primero permite adaptar la grilla de recursos para la utilización de minislots (se explica en el Sección 7.1) y el segundo se utiliza para definir el slot según 5G (pues el simulador lo traía definido por defecto para LTE). El subpaquete “+settings” recopila y define las opciones de configuración que se pueden realizar en una simulación. Se puede elegir simulación de downlink o uplink (hasta el momento sólo es posible downlink), tipo de simulación (local o paralela), tipo de scheduler (Round Robin, BestCQI, Proportional Fair (ver Capítulo 6) o framework (ver Capítulo 8)), tipo de grilla, modelo de pérdidas, interferencia, tipo de feedback entre otros.

Parámetros de scheduling

El scheduling se realiza utilizando como unidad de tiempo base el TS, al cual es posible asignarle un valor libremente, por ejemplo se le puede dar el valor de 1 milisegundo para representar subtramas LTE-A. Otra unidad de tiempo utilizada en el simulador es el “segmento”: el cual se define como el lapso de tiempo durante el cual los valores de MF (Macroscopic Fading) son aproximadamente constantes, por lo tanto estos valores se actualizan una vez por segmento. Una secuencia de segmentos consecutivos se define como “chunk”. De esta manera la asignación a una BS basada en la potencia de señal recibida o SINR, no se modifica durante en transcurso del mismo segmento [13]. Dentro del paquete “+parameters” se encuentra el archivo “Time.m” donde se inicializan los valores de cantidad de chunks, duración del TS, cantidad de slots por chunk y tiempo entre chunks. A partir de esto se inicializa la matriz de tiempos en la que se basa la simulación.

En la etapa de inicialización se determina la longitud del segmento en función de la velocidad, la trayectoria de los usuarios y la distancia de descorrelación especificada. Para usuarios estacionarios se crea un solo segmento y por ello los valores de MF se calculan una única vez. Entre chunks transcurre un tiempo considerable (se fija mediante el parámetro “timeBetweenChunksInSlots”), de modo que es posible que cambie significativamente la posición de los usuarios. Indepen-

Capítulo 4. Simulador Vienna 5G SLS

dientemente de la movilidad, se supone que el entorno de dispersión cambia la señal entre chunks de modo que los coeficientes de canal no están correlacionados. A partir de esta independencia entre chunks es posible realizar todos los cálculos y los resultados generados dependen únicamente del chunk en cuestión. De esta manera es posible separarlos y procesarlos en paralelo.

Los parámetros no especificados en el escenario de simulación son fijados con un valor por defecto. Además de la definición de los parámetros se debe elegir alguno de los escenarios predefinidos con los que cuenta el simulador.

Elección del escenario

El paquete “+scenarios” del simulador contiene ocho escenarios predefinidos y dos escenarios implementados para el proyecto (además otros que son variantes de los dos últimos). En cada uno de los escenarios se inicializa, en primer lugar, la configuración de tiempos: números de chunks por simulación, duración del slot en segundos, cantidad de slots por chunks, tiempo entre chunks (medido en slots) y retardo en la realimentación. También es posible definir la RoI (Region of Interest) y la región de interferencia, ambas representadas tridimensionalmente (con las coordenadas x,y,z en metros). Se le asignan valores a los parámetros que modelan las antenas (tipo de antena, cantidad de antenas transmisoras y receptoras, potencia, altura de la antena). Se distribuyen las BSs y los usuarios mediante un factor de densidad y la posición de los mismos se modela con una distribución de Poisson. Si el escenario incluye edificios los mismos son modelados según su altura, largo, ancho. Se indica también el ancho de las calles, la pérdida que introducen las paredes y los paredones que separan los edificios de la calle.

Se presenta a continuación la lista de los escenarios predefinidos que contiene el simulador:

- **basicScenario:** Escenario sin edificaciones, con las BS distribuidas en grilla hexagonal. Incluye únicamente usuarios eMBB ubicados espacialmente con distribución Poisson. El post procesamiento se realiza en modo “PartialPP” lo que significa que se guardan los resultados intermedios en la estructura “SaveObject”.
- **basicLiteScenario:** Escenario con usuarios y BS ubicados de forma aleatoria mediante distribución Poisson. En una simulación lite (o ligera) el post procesamiento se puede realizar en modo “LiteNoNetworkPP” o “LiteWithNetworkPP”. El primer modo no guarda información de la red, el segundo modo se utiliza para simulaciones donde no interesa el LQM (Link Quality Monitoring) pero sí interesa la geometría de la red.
- **HetNet:** Escenario heterogéneo con varias BS y tipos de usuarios. Se tienen en cuenta macro, pico y femto BS con usuarios peatones o en vehículos.
- **hexRingInterferers:** En este escenario se ubica un número “nRing” de anillos de BS y un número “nRingInterferers” de anillos interferentes de forma de grilla hexagonal. Cada BS está equipada con antenas de tres sectores.

4.1. Estructura del simulador

- `interferenceRegionPPP`: En este escenario se distribuyen (en la RoI) las BS con una región de interferencia para mitigar los efectos de borde. Los usuarios y las BS están distribuidas según un PPP (Poisson Point Process).
- `IoTclusteredUser`: Este escenario genera un gran número de usuarios en cluster con las BS distribuidas en PPP en una red con dos niveles. Es útil para mostrar simulaciones con muchos usuarios con actividad en intervalos regulares de tiempo.
- `ManhattanGridScenario`: Este escenario se configura para una simulación sobre una cuadrícula del estilo Manhattan con calles y edificios. Las BS se colocan encima de los edificios y los usuarios son distribuidos con PPP a nivel del suelo.
- `UserMovement`: Escenario simple (basado en el `basicScenario`) con usuarios con movimiento.

Además se crearon dos escenarios con características relacionadas a 5G para ejecutar las simulaciones y probar las implementaciones realizadas:

- `NR`: Es un escenario basado en el `basicScenario` (con dimensiones espaciales mayores) pero también incluye movimiento en los usuarios (se estudian solo usuarios del tipo eMBB). Se le agregaron los campos “`trafficModelType`”, “`trafficLimit`” y “`userType`” que son necesarios para indicar las características del modelo de tráfico implementado (ver Capítulo 5) y el tipo de usuario que se estudia.
- `NR_Minislot`: Este escenario se basa en el escenario NR y se utiliza para simulaciones con usuarios de eMBB (tipo 1) conviviendo con usuarios de baja latencia o URLLC (tipo 2). Por esta razón se agregaron los campos con las características que describen a los dos tipos de usuarios.

Chequeo de compatibilidad

Para finalizar la inicialización se debe comprobar la compatibilidad de las características configuradas en los parámetros para evitar problemas o inconsistencias durante la simulación. En caso de aparecer algún parámetro inusual se advierte o se devuelve un error en caso de parámetros imposibles. El chequeo de compatibilidad se realiza sobre la RoI. A modo de ejemplo: se verifica que la altura de la zona de interferencia no sea mayor que la altura de la RoI, al igual que las BS o la posición de los usuarios no sea mayor que ésta.

4.1.2. Pregeneración

Otro de los bloques en los que se divide el simulador es el de pregeneración. Este es el encargado de crear todos los elementos con posición geométrica especificadas en R^3 . Existe la posibilidad de descartar la tercera dimensión para realizar simulaciones más ligeras en cuanto a costo computacional. A continuación se describen los elementos de la red que se generan.

Capítulo 4. Simulador Vienna 5G SLS

Elementos de la Red

En el paquete “+networkElements” se define una superclase que abarca los usuarios y antenas con su ubicación en la RoI. En el archivo “NetworkElement-WithPosition” se encuentran las funciones que asignan las posiciones de los elementos en la red. También contiene una función para calcular la potencia del ruido térmico. Este paquete cuenta con los subpaquetes “+bs”, “+quick”, “+ue”.

En el subpaquete “+bs” están definidos los tipos de antenas: omnidireccional, de un sector, de tres sectores, de seis sectores y de tres sectores tipo Berger. También se definen los tipos de BS que pueden tener asociadas varias antenas diferentes. Además con esta clase es posible implementar RRH (Remote Radio Heads) o DAS (Distributed Antenna System). El subpaquete “+quick” configura los elementos para un escenario básico de simulación.

El subpaquete “+ue” define la clase para un usuario receptor en el enlace de bajada. Se encuentran definidas propiedades como el “id”, un entero que representa el número de identificación para ese usuario y el “nRX” junto con el “nTX” que es el número de antenas receptoras y transmisoras de ese usuario. Se define el modelo de canal para el usuario y la velocidad a la que se desplaza. Otro parámetro que se define aquí es el correspondiente al modelo de tráfico, que fue implementado por el equipo ya que el simulador funcionaba en modalidad de full buffer¹ (lo que significó una limitación importante). Con esta implementación fue posible analizar más en profundidad el comportamiento de los usuarios en las simulaciones ya que permitió variar el tráfico requerido y corroborar si la demanda de datos fueron cubiertas o no (esto se detalla en el Capítulo 5). También se agregó aquí el campo “UserType” con el cual es posible caracterizar al usuario como eMBB o URLLC (dependiendo el servicio que se quiera estudiar). Originalmente el simulador trabajaba solamente con usuarios de servicio de banda ancha (eMBB).

Generación

En primera instancia se generan las edificaciones y muros (generación de edificios) las cuales son colocadas en el área de simulación de acuerdo a lo especificado en el escenario. Luego son generadas las BS siguiendo los parámetros especificados. Las antenas asociadas a las BS tienen una ubicación específica lo que podría permitir, por ejemplo, simular sistemas de antenas distribuidas. Por último cuando todos los elementos fueron generados se establece la configuración de cada chunk de simulación incluyendo dichos elementos (distribución en chunks). Estos ajustes son los correspondientes a cada chunk permitiendo reducir el overhead de información durante la simulación y permitiendo la simulación en paralelo.

4.1.3. Bucle de Simulación Principal

Una vez generada la configuración por chunk se procede al bucle principal de simulación (que se ilustra en la Figura 4.2) donde se procesan los chunks. Cada

¹En la modalidad tipo “full buffer” se tiene a cada usuario de la radiobase en cuestión solicitando todos los recursos disponibles.

4.1. Estructura del simulador

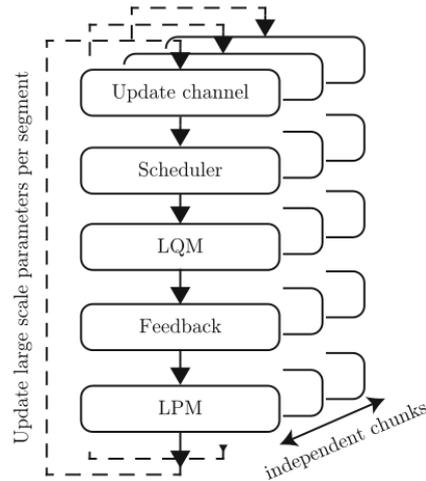


Figura 4.2: Bucle de simulación de un chunk sobre los time slots [13].

simulación de chunk consiste en un bucle que se ejecuta sobre cada TS.

Al principio de cada slot se verifica que no se encuentre un segmento nuevo (en caso de considerarse que hubo un nuevo segmento, se procede a calcular las nuevas posiciones de los usuarios con movimiento y se calculan también los nuevos parámetros que son dependientes de la posición como lo son el path loss, BS asignada, la ganancia de la antena y la condición del canal). En el paquete “+simulation” se encuentra en script “ChunkSimulation.m” el cual se ejecuta en cada chunk que transcurre. Entre otras funciones que cumple inicializa el scheduler recorriendo las BS y posteriormente realiza un bucle por todos los slots del chunk actualizando la RoI, creando resultados temporales, guardándolos y realizando el scheduling (entre otras actualizaciones de parámetros que realiza). Dentro del paquete “+simulation” también se encuentran los scripts “LocalSimulation.m” (ejecuta la simulación en la máquina local) y “ParallelSimulation.m” (realiza una simulación paralelizada, requiere el Parallel Computing Toolbox de Matlab) que son la parte central de la simulación (se utiliza uno u otro dependiendo el modo en que se elija ejecutar la simulación). Dentro de los mismos se llama a la función “ChunkSimulation”. En la figura 4.3 se representa, a modo de ejemplo, la relación entre LocalSimulation, ChunkSimulation y RoundRobinScheduler (podría ser cualquiera de los schedulers).

El resto se puede dividir en 5 pasos que se muestran en la Figura 4.2: primero se actualiza el canal por medio del channel model, luego se realiza el scheduling que se encarga de asignar los recursos tanto de espectro como de potencia en cada una de las BS (las funciones y archivos relacionados a este paso se encuentran especificados en la Sección 4.1.3). Posteriormente se produce la etapa LQM (Link Quality Model) donde se combina la codificación, modulación y caracterización del canal obteniendo un valor de SINR post ecualización (este valor es el que se utiliza para representar el desempeño de la capa física). En el siguiente paso se realiza el cálculo del feedback que puede ser, por ejemplo, el CSI de LTE-

Capítulo 4. Simulador Vienna 5G SLS

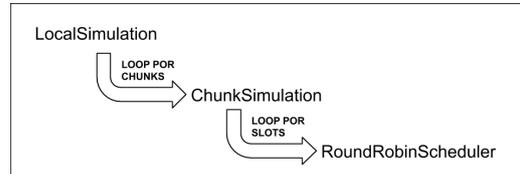


Figura 4.3: Bucle de simulación implementado en el Vienna 5G SLS.

A compuesto por el valor del CQI, el RI (Rank indicator) y el PMI (Precoding Matrix Indicator). El feedback debe estar alineado con el scheduler utilizado ya que recibe esta información y podría utilizarla para calcular métricas. Luego en el último paso del bucle se realiza el LPM (Link Performance Model) que a partir del SINR (post ecualización) calcula el BLER (Block Error Rate) y el throughput teniendo en cuenta el MCS utilizado.

A continuación se describe en mayor profundidad el paquete “+scheduler” (que contiene los schedulers del simulador) ya que será la sección más trabajada en nuestro proyecto.

Schedulers

El paquete “+scheduler” del simulador contiene los scripts correspondientes a los schedulers Round Robin (“RoundRobinScheduler.m”) y Best CQI (“BestCQIScheduler.m”) que existían por defecto en la versión original del simulador. Además contiene dos schedulers adicionales: uno basado en el algoritmo Proportional Fair (“PropFairScheduler.m”) y otro que se corresponde al Framework (“FrameworkScheduler.m”). El primero (Proportional Fair) atiende a usuarios eMBB (ver Capítulo 6) y agrega la técnica de puncturing para atender a usuarios URLLC (ver Sección 7.2). El segundo (Framework) fue implementado para elaborar algoritmos personalizables basados en métrica a partir de ciertos parámetros de la red (ver Capítulo 8). Tanto en el script correspondiente al Round Robin como en el Best CQI se hicieron modificaciones adicionales para implementar el modelo de tráfico (ver Capítulo 5) ya que por defecto los schedulers funcionan en modo full buffer.

También dentro del paquete “+scheduler” se encuentra el archivo “Scheduler.m” que contiene la definición de la clase “Scheduler” de la cual se heredan los atributos y funciones de cada uno de los schedulers (se dice que es una clase “padre”). En este caso se define la clase “RoundRobinScheduler” (que es hija de la clase “Scheduler”) y los demás schedulers (Best CQI, Proportional Fair y Framework) son hijos de la clase “RoundRobinScheduler” (esto se ilustra en la Figura 4.4).

En este archivo (“Scheduler.m”) se encuentran definidos los métodos “scheduleDL”, “addUserDL” y “removeUserDL”. El primer método (“scheduleDL”) es la parte central del algoritmo de scheduling que aparece definido como función en cada uno de los scripts de los schedulers (“RoundRobinScheduler.m”, “BestCQIScheduler.m”, “PropFairScheduler.m” y “FrameworkScheduler.m”). Los dos últimos (“addUserDL” y “removeUserDL”) se encargan de agregar/quitar usuarios a/desde la cola directamente, lo que es eficiente cuando se produce el handover. La actua-

4.1. Estructura del simulador

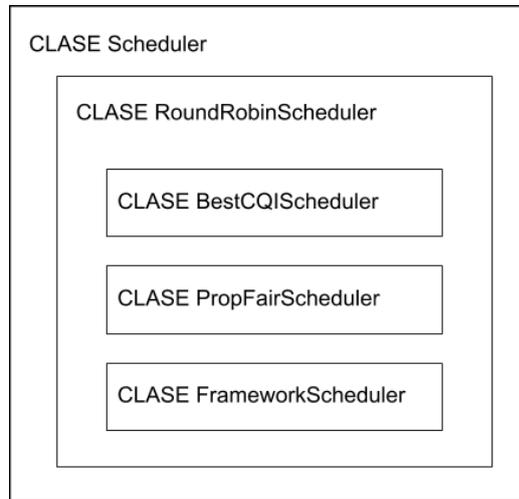


Figura 4.4: Representación de la jerarquía de clases de los schedulers.

lización de los usuarios que se encuentran enlazados a la BS se realiza a través de la función “updateAttachedUsersDL” que llama a las funciones “addUserDL” y “removeUserDL”.

Otras funciones que se encuentran definidas en “Scheduler.m” son:

- “scheduleDLCommon”: Se define la información que es común en todos los tipos de schedulers, por ejemplo aquí se actualiza el valor del tmodel para cada usuario.
- “getOptimumCQI”: Busca el valor más alto de CQI que supera cierto umbral de BLER (dicho valor es el mismo para todos los RB). Este CQI promedio se calcula para cada codeword.
- “generateScheduler”: Llama al constructor del scheduler que está dado en “config.schedulerParameters.type”.
- “getTBSIZEbits”: Calcula el número de bits de datos que contienen los RB que fueron asignados a un usuario.

El paquete “+scheduler” contiene el archivo “SchedulerSignaling.m”, en el mismo está definida la clase correspondiente a la información generada para cada usuario. También contiene el archivo “rbGrid.m” en el cual se definen los siguientes campos relacionados a la grilla de recursos:

- “userAllocation”: Es el índice del usuario al cual le asignaron el recurso (tiene dimensiones de cantidad de RB en frecuencia por cantidad de RB en el tiempo).
- “powerAllocation”: Es la potencia asociada a los datos asignados en el recurso (tiene dimensiones de cantidad de RB en frecuencia por cantidad de RB en el tiempo).

Capítulo 4. Simulador Vienna 5G SLS

- “powerAllocationSignaling”: Es la potencia de la señalización asociada al RB asignado (tiene dimensiones de cantidad de RB en frecuencia por cantidad de RB en el tiempo).
- “CQI”: Es el CQI usado por cada RB y codeword (tiene dimensiones de cantidad de RB en frecuencia por cantidad de RB en el tiempo por número de codewords).
- “precoder”: contiene las matrices de precodificación (tiene dimensiones de cantidad de RB en frecuencia por cantidad de RB en el tiempo).
- “nLayers” y “nCodewords”: Es el número de layer y codewords utilizado (tienen dimensiones de cantidad de RB en frecuencia por cantidad de RB en el tiempo).

4.1.4. Post procesamiento

En esta última etapa se combinan todos los resultados individuales de cada usuario por BS y en cada chunk (luego de extraer la información de cada slot). Dentro del paquete “+simulation” se encuentran dos subpaquetes: “+results” y “+Postprocessing”. En el primero están definidas todas las clases donde se definen las funciones que extraen y grafican datos temporales y parciales. En el segundo subpaquete se define la clase “PostprocessorSuperclass” donde están las funciones que se encargan de crear una única estructura con los resultados. Esta estructura se muestra al final de la simulación y se guarda un archivo con estos datos. El post procesador predefinido (que usan todos los escenarios) se llama PartialPP y se encarga de rellenar la estructura “trace” con los resultados obtenidos en cada slot (“temporaryResult”).

La clase “Result” es la que contiene las funciones encargadas de graficar las posiciones de los usuarios, antenas y obstáculos (también el movimiento de los usuarios si se desea) que se muestran al finalizar la simulación. Estas funciones son llamadas en la clase “ResultsSuperclass” donde también están implementadas las funciones que grafican el throughput en Mbit/s por usuario (que se conoce como ECDF: Empirical Cumulative Distribution).

4.2. Limitantes a destacar del Simulador Vienna 5G SLS

El simulador no maneja información de HARQ lo que impacta en los siguientes aspectos: en la elección del MCS se utiliza solo el CQI (sin información de BLER), con lo cual se puede elegir un valor que lleve a muchas retransmisiones. Un ejemplo de esto es el caso donde se tiene una sección de espectro con más interferencia que el resto. El CQI se promedia en toda la banda, en general el móvil reporta un valor de CQI para la banda (por más que el Vienna ponga uno por PRB), lo que levanta el promedio y no refleja fielmente un bajo SINR por interferencia en cierta zona del espectro. El algoritmo de scheduling en base al CQI promedio puede no dar buenos resultados si se asignan PRBs en zona interferida.

4.3. Resumen del Capítulo

Otro aspecto a tener en cuenta es que cada retransmisión debe manejarse como parte de la información a asignar por el scheduler y usualmente se le da un peso mayor a este tráfico frente a transmisiones iniciales. Considerando que el MCS se elige para un BLER target del orden de 10%, esta cantidad de retransmisiones no es despreciable.

4.3. Resumen del Capítulo

En el presente capítulo se explicó la estructura del simulador Vienna 5G SLS y además se expusieron los bloques que fueron modificados para realizar las diferentes implementaciones. Se logró describir de forma general el funcionamiento del simulador y sus partes más representativas.

A continuación se desarrollará sobre la implementación y validación del modelo de tráfico realizado.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Modelo de tráfico

Dada la versión 1.0 en la que se encontraba el Vienna 5G SLS, el simulador no contaba con modelo de tráfico, el simulador funcionaba en modalidad full buffer. Dadas las grandes limitantes que esto supone para el estudio de distintos algoritmos de scheduling, fue necesario agregar la funcionalidad de limitar el tráfico transmitido de acuerdo a un modelo de tráfico.

5.1. Modelo de tráfico requerido por los usuarios

La versión original del simulador contenía implementados los schedulers Round Robin y Best CQI, como se mencionó anteriormente, ambos funcionando en full buffer. Se entiende por full buffer que cada usuario no cuenta con una limitación en el tráfico requerido, es decir, se asignarán todos los recursos suponiendo que todos los usuarios quieren continuar transmitiendo. Este funcionamiento no se adapta a lo que sucede en la realidad, puesto que para cada usuario, dependiendo el servicio que utilice, es diferente el volumen de tráfico que necesita y por consiguiente los recursos que debe asignar la radiobase. Para introducir el estudio de diferentes usuarios con diferentes servicios se decidió implementar un modelo de tráfico para los usuarios y de esta forma extender las funcionalidades del simulador.

Se decide que para cada usuario se almacene la cantidad de tráfico a transmitir, utilizando la estructura temporal del simulador de división en chunks y slots. A cada usuario se le asigna una matriz de tamaño número de chunks por número de slots, dicha matriz contiene la cantidad de tráfico que requerirá el usuario (medido en bits) en cada slot. Por ejemplo si la simulación tuviera dos chunks de seis slots cada uno la matriz podría tener el aspecto mostrado en la matriz 5.1. La matriz del ejemplo debe interpretarse de la siguiente forma: la primer fila que representa el primer chunk, en el slot uno el usuario requiere 1000 bits, en el slot dos del primer chunk el usuario requiere 1300 bits, así sucesivamente con toda la fila. De forma análoga en la segunda fila (que representa el segundo chunk) el 900 representa que en el primer slot el usuario requiere 900 bits, el 0 indica que en el segundo slot el usuario no requiere bits y así hasta el final de la fila. Dicha matriz es actualizada luego de cada transmisión, de esta manera si los bits solicitados no pueden ser

Capítulo 5. Modelo de tráfico

satisfechos lo que resta por transmitir es sumado en el siguiente slot.

$$\begin{pmatrix} 1000 & 1300 & 120 & 0 & 700 & 1400 \\ 900 & 0 & 1300 & 1400 & 600 & 900 \end{pmatrix} \quad (5.1)$$

La limitación de tráfico dentro del scheduler debe hacerse de acuerdo a la cantidad de bits a transmitir en cada recurso. De la misma forma la actualización de los valores de la matriz debe tener en cuenta la cantidad de bits de los recursos asignados en lugar de la cantidad de recursos.

Se diseñó también un modelo de tráfico para los usuarios URLLC (servicio explicado en el Capítulo 2). En este caso se agrega una tercera dimensión a la matriz mencionada anteriormente que representa los minislots correspondientes a cada slot. El valor contenido en la misma corresponde a la cantidad de fragmentos de RB (ya que será de largo 1 minislot en lugar de 1 slot) que requerirá en cada minislot. La forma de rellenar la matriz serán la misma que para los usuarios eMBB pero la actualización de la matriz será por minislot (pasando el déficit, en caso de haber, al siguiente minislot). En este caso no se utiliza como unidad el bit dado que los usuarios de baja latencia requieren alta confiabilidad por lo que el esquema de modulación elegido de acuerdo al CQI no será el mismo (ver tablas D.3 y D.1 en Apéndice D). Por lo tanto la cantidad de bits a enviar en cada recurso no será la misma que si fuera un usuario eMBB, entonces esto provocaría incongruencias con los resultados del throughput de los usuarios de baja latencia. De todas formas el objetivo de la adición de usuarios de baja latencia es estudiar la influencia sobre el resto del sistema, por lo que carece de importancia contar con un cálculo preciso de la cantidad de bits transmitidos.

Para acoplar el modelo de tráfico en el simulador fue necesario modificar varios objetos existentes (por ejemplo el objeto “user” correspondiente al usuario). De esta forma se pudo: almacenar la matriz para cada usuario, asignar recursos de acuerdo a la cantidad especificada y actualizar el modelo de forma adecuada. A continuación se explica la forma en que se adaptó el simulador para que funcione con el modelo de tráfico.

5.1.1. Adaptación del simulador al modelo de tráfico

Para agregar la información del tráfico requerido por cada usuario se implementa el uso de un campo adicional en el objeto usuario: el parámetro “tmodel”. Este campo es una matriz de tamaño número de chunks por número de slots que se rellena de forma estadística durante la creación de los usuarios en el escenario.

El modelo de tráfico inicial es asignado a los usuarios al principio de la simulación y los valores del mismo varían a lo largo de los slots (adaptándose mejor a la realidad). De forma de agilizar el proceso de simulación se incluyeron (en el modelo de tráfico) diferentes funciones para rellenar de forma estadística las matrices. Los modelos incluidos son:

- Constante (“const”): El usuario requiere la misma cantidad de bits en todos los slots. Dicho valor es especificado mediante el campo “trafficLimit”.

5.1. Modelo de tráfico requerido por los usuarios

- Aleatorio (“randi”): En todos los slots se requiere una cantidad aleatoria de bits entre 0 y el valor “trafficLimit”. Se utiliza distribución Uniforme.
- Ráfagas (“burst”): Se elige de forma aleatoria (binomial) si un usuario transmite o no en cada slot. En caso de transmitir el valor será “trafficLimit”.

Dada la abstracción del modelo, es muy simple la adición de nuevas formas de distribuir inicialmente el tráfico. La implementación de este modelo de tráfico (requerido por los usuarios) es muy intuitiva: el scheduler asigna recursos a los usuarios mientras los necesiten (necesita si “tmodel” del usuario en ese slot mayor a cero) y existan recursos disponibles (matriz “userAllocation” del simulador no llena). Al asignar un recurso se calcula cuál es la capacidad en bits de dicho recurso para el usuario al que se le asignó (de acuerdo al CQI del usuario) y se lleva una cuenta de cuántos bits transmitirá cada usuario en el slot actual. En el caso de que se terminen los recursos disponibles y existan usuarios que aún requieran recursos (déficit) entonces se suman los bits del déficit en el siguiente slot. El déficit en el último slot del chunk no se toma en cuenta (queda sin saldar).

Además al terminar la asignación en el slot se ejecuta la función “scheduleDL-Common” (definida en el archivo “Scheduler” explicado en Sección 4.1.3) donde se calcula cuál es el tamaño de transport block a transmitir por cada usuario. Aquí se calcula el déficit de bits y (en caso de existir) se suma al valor de tráfico del siguiente slot. Dado que el cálculo de bits que posee el simulador estaba pensado para obtener el tamaño de un TB (Transport Block) entero y no de recursos individuales (necesario para saber recurso a recurso cuántos bits se asignaron) se debió modificar la función que realiza el cálculo. Para esto fue necesario conocer los bits de información de datos que contienen los recursos asignados a cada usuario.

El simulador ya contaba con una función para el cálculo de bits contenidos en un transport block, se hacía de forma idéntica al caso de LTE utilizando la línea

```
tbSizeBits = nlayers*nDataSymbols*modulationOrder*codingRate)-  
nCRCBits
```

donde:

- “modulationOrder” (orden de modulación) y “codingRate” (tasa de codificación) se obtienen de la Tabla D.2 compartida entre LTE y 5G NR [23] (ver Apéndice D)
- “nCRCBits” es la cantidad de bits utilizados para el CRC (Cyclic Redundancy Checksum) que es configurada para LTE en 24
- “nlayers” es la cantidad de layers a utilizar en la transmisión (es obtenida a partir de feedback)
- “ndataSymbols” es la cantidad de símbolos de datos teniendo en cuenta que en algunos recursos se envían símbolos de referencia y sincronización

Ese cálculo no se adecuaba a un cálculo de bits transmitidos en recursos individuales (necesario para control del tráfico por parte del scheduler). Usar una

Capítulo 5. Modelo de tráfico

estimación diferente generaría incongruencias entre lo asignado por el scheduler y lo alojado realmente en los recursos por lo que se decidió modificar dicho cálculo y utilizar la misma función durante la asignación de los recursos y al calcular los bits a transmitir luego de haber asignado todos los recursos.

La modificación se basó en la Ecuación 5.2 (extraída del release 15 de la 3GPP [50]):

$$data\ rate(inMbps) = 10^{-6} \cdot \sum_{j=1}^J \left(v_{Layers}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{max} \cdot \frac{N_{PRB}^{BW^{(j)} \cdot \mu} \cdot 12}{T_s^\mu} \cdot (1 - OH^{(j)}) \right) \quad (5.2)$$

En la Ecuación 5.2:

- J: es el número de portadoras en una banda o combinación de bandas
- $R_{max} = 948/1024$, para el j-ésimo CC (Carrier Component)
- $v_{Layers}^{(j)}$ es el número máximo de layers
- $Q_m^{(j)}$ es el orden máximo de modulación
- $f^{(j)}$ es el factor de escala (que puede tomar valores de 1, 0.8, 0.75 y 0.4),
- μ es la numerología
- T_s^μ es el promedio de la duración de los símbolos OFDM en un subframe de la numerología μ
- $N_{PRB}^{BW^{(j)} \cdot \mu}$ es la máxima cantidad de RBs en el ancho de banda $BW^{(j)}$ con numerología μ
- $OH^{(j)}$ es el overhead que puede tomar los valores 0.14 para el rango de frecuencia RF1 y 0.18 para el rango de frecuencia RF2 (ambos en DL)

Como se puede observar dicha ecuación corresponde a la capacidad total del canal OFDM, pero lo que interesa es el overhead estimado que se utiliza. Con el objetivo de simplificar el cálculo de la cantidad de bits y que todos los recursos sean afectados de igual forma por el overhead se propone el método

$$tbSizeBits = nlayers * 8 * \text{round}((1/8) * nDataSymbols * modulationOrder * codingRate * (1 - overhead))$$

donde “nDataSymbols” no tiene en cuenta los símbolos de sincronización sino que es calculado como indica la Ecuación 5.3.

$$nDataSymbols = 12 \cdot nSymbolRb \cdot nAssignedRBs \quad (5.3)$$

En la Ecuación 5.3:

- 12 es el número de subportadoras por PRB

5.2. Validación del modelo de tráfico

- “nSymbolRb” es la cantidad de símbolos por recurso
- “nAssignedRBs” es la cantidad de recursos asignados (cuando la función es utilizada para recursos individuales toma valor 1)

De esta forma se simplifica el cálculo de recursos individuales y el resultado global de la capacidad de la portadora es el mismo que el calculado en la recomendación de la 3GPP [50].

5.2. Validación del modelo de tráfico

A lo largo de esta sección se muestra la validación que se realizó sobre el modelo de tráfico. El mismo debe cumplir dos reglas básicas que son las que se verificaron:

- En caso de no ser posible cubrir el valor requerido de bits (quedando en déficit) el faltante se debe adicionar en el siguiente slot (en caso de existir).
- La asignación de recursos debe ser la menor cantidad tal que los requisitos solicitados en el modelo de tráfico sean satisfechos.

Para validar dicho comportamiento se realizaron simulaciones con distintos parámetros, a continuación se detalla una de ellas y se analiza el comportamiento. En la Tabla 5.1 se muestra un resumen de las características del escenario utilizado. Para realizar un análisis detallado se decidió diseñar un escenario simple con una sola radiobase y un número reducido de usuarios (en el ejemplo a analizar existen cinco usuarios estáticos y uno en movimiento). Las condiciones del medio son simples para aislar lo mayor posible el scheduler y el modelo de tráfico del resto del sistema.

Al correr una simulación con el escenario planteado anteriormente, la posición de los elementos de la red es la mostrada en la Figura 5.1, donde el punto rojo en el centro es la radiobase y el resto los usuarios.

A los resultados de la simulación planteada previamente, se le realizó un procesamiento de forma tal de obtener gráficos que muestren resultados pertinentes para la validación. Los valores que interesa visualizar son: la cantidad de recursos asignados a cada usuario y la cantidad de bits que pudieron ser transmitidos en esos recursos. Dichos resultados son contrastados con el modelo de tráfico configurado para verificar las condiciones que debe cumplir el sistema de control de tráfico.

En la Figura 5.2 se muestran los bits enviados y en la Figura 5.3 los recursos asignados por cada usuario en cada slot. Dado que en el primer slot no se tiene feedback, todos los usuarios se asumen con $CQI = 1$ por lo que los bits transmitidos son pocos a pesar de haberse asignado todos los recursos. Debido a esto, en el primer slot, todos los usuarios tienen un gran déficit de datos enviados dado que solicitaron mucho más de lo que se les asigna. Dicho déficit es cubierto en los siguientes slots, como se puede observar el total de recursos asignados es “106” hasta el slot “9”. 106 es el total de recursos disponibles del escenario.

Capítulo 5. Modelo de tráfico

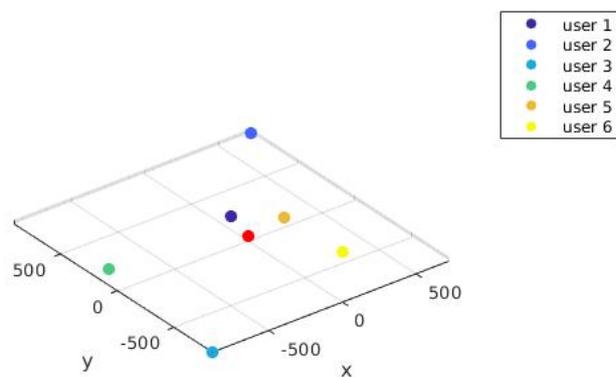


Figura 5.1: Gráfico de elementos de la red de simulación para validación de modelo de tráfico.

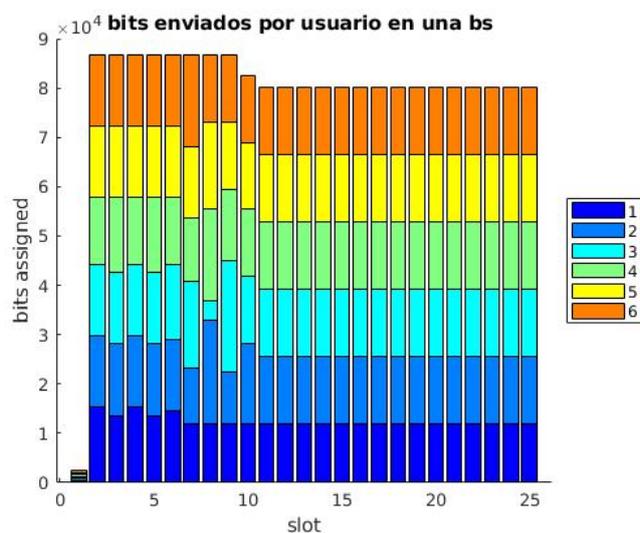


Figura 5.2: Gráfico de bits enviados por cada usuario en la radiobase utilizada para validar el modelo de tráfico.

5.2. Validación del modelo de tráfico

Tabla 5.1: Tabla resumen de escenario de simulación para validación de modelo de tráfico

Espacio			
Dimensiones	x=1900	y=1900	z=100
Obstáculos	Espacio abierto		
Tiempo			
slots	25		
feedbackDelay	1		
Transmisión			
Ancho de banda útil para transmisión	98 %		
Overhead	14 %		
Ancho de banda de portadora	20MHz		
Duración del slot	1 ms		
Scheduler	Proportional Fair		
Frecuencia central	2 GHz		
Radiobases			
Tipo	Macro		
Cantidad	1		
Antena	Omnidireccional		
Usuarios			
Distribución espacial	Poisson		
Densidad usuarios en movimiento	0.00000002		
Trafico de usuarios en movimiento	Constante 12000 bits por slot		
Velocidad	50 km/h		
Densidad usuarios estáticos	0.0000015		
Trafico de usuarios estáticos	Constante 13000 bits por slot		

5.2.1. Verificación de traspaso de déficit

Para mostrar que el déficit de bits enviados es cubierto en los siguientes slots se tomó uno de los usuarios y se graficaron los valores de tráfico solicitado, tráfico solicitado originalmente (sin incluir déficit previo), bits enviados y déficit acumulado luego del slot. Dicho gráfico se observa en la Figura 5.4. En la misma se observa que la barra de déficit acumulado fue muy similar a la de bits solicitados en el primer slot ya que se enviaron pocos bits. Luego hasta el slot 6 se enviaron más bits de los solicitados originalmente provocando un decrecimiento del déficit. En el slot 7 no fue posible cubrir el tráfico solicitado originalmente debido a que se le asignó mayor cantidad de recursos al usuario 6 (dicho comportamiento se observa

Capítulo 5. Modelo de tráfico

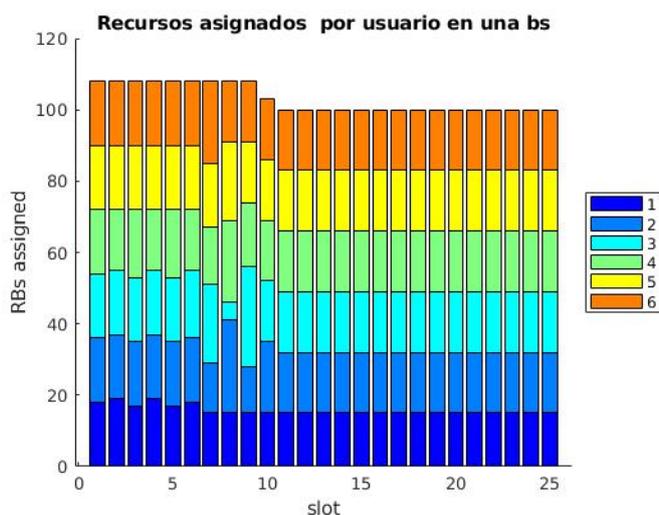


Figura 5.3: Gráfico de recursos asignados por cada usuario en la radiobase utilizada para validar el modelo de tráfico.

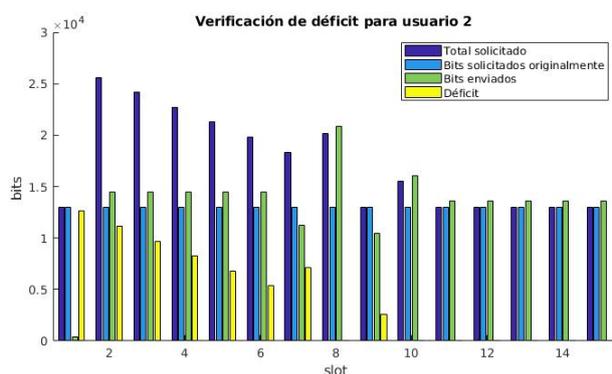


Figura 5.4: Gráfico de recursos asignados, déficit y tráfico requerido por usuario 2 en la radiobase utilizada para validar el modelo de tráfico.

en la Figura 5.3), por esta razón el déficit vuelve a crecer. En cambio en el slot 8 la asignación de recursos para dicho usuario creció considerablemente (como se observa en la figura de recursos asignados) por lo que todo el déficit fue cubierto. En el siguiente slot se puede observar nuevamente un déficit pero el mismo es cubierto en el slot 10. En los siguientes slots (11 en adelante) se mantiene invariante la asignación sin déficit.

5.2.2. Verificación de asignación mínima

El otro requisito a cumplir por el modelo de tráfico dentro del scheduler es que la asignación de recursos sea la mínima necesaria para cubrir los requisitos. En la Figura 5.5 se muestra la cantidad de bits sobrantes por slot (en caso de existir) y la cantidad de bits enviados por recurso. El valor de excedente de bits debe ser menor

5.3. Resumen del Capítulo

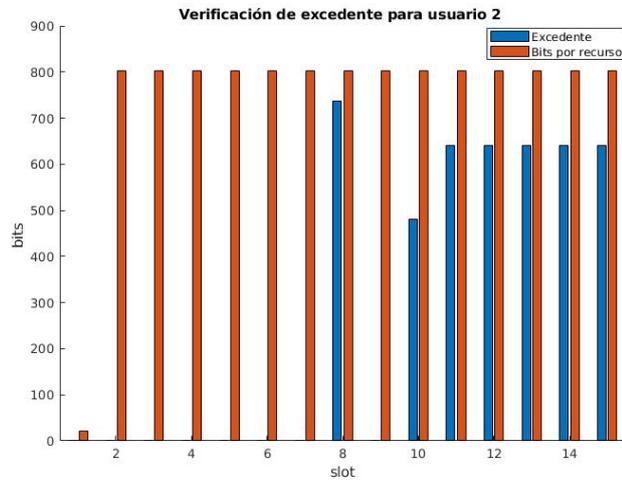


Figura 5.5: Gráfico de excedente de bits asignados y cantidad de bits por recurso del usuario 2 en la radiobase utilizada para validar el modelo de tráfico.

estricto que la cantidad de bits por recurso (dado que si es mayor quiere decir que podría haberse asignado uno o más recursos y aún así satisfacer los requisitos de tráfico).

Como se puede observar el excedente siempre es menor a la cantidad de bits por recurso. Por lo tanto se concluye que la asignación es mínima.

5.3. Resumen del Capítulo

En el presente capítulo se detalló la implementación y validación del modelo de tráfico que permitió definir requerimientos de tráfico para los usuarios. De este modo se amplió el funcionamiento del simulador que originalmente trabajaba solo en modalidad full buffer.

A continuación se desarrolla la implementación y validación del algoritmo Proportional Fair Generalizado (o con pesos).

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Proportional Fair Generalizado

Uno de los objetivos planteados del proyecto fue crear algoritmos de scheduling diferentes a los ya existentes en el simulador (BestCQI y Round Robin, ambos full buffer). Luego de estudiar varias opciones se decidió hacer un algoritmo basado en el Proportional Fair (definido en el Capítulo 3) cuyo uso es bastante extendido en la tecnología LTE. En el presente capítulo se detallan las características y el funcionamiento del algoritmo diseñado, además de las diferencias con otras versiones del mismo algoritmo.

6.1. Diseño del algoritmo

El algoritmo implementado es un Proportional Fair generalizado (o con pesos). Este algoritmo permite mantener el balance deseado entre repartir los recursos de forma equitativa y maximizar el throughput total de la radiobase, permitiendo a su vez un ajuste del balance mediante la asignación de pesos. El diseño de este algoritmo se basa en un Proportional Fair [51] al que se le hacen algunas modificaciones para poder flexibilizarlo. El algoritmo consta en asignar los recursos (RBs) a los usuarios priorizando de acuerdo a una métrica. Cada usuario posee una métrica para cada recurso. En el momento de la asignación, se le dará cada recurso al usuario con mejor métrica. Esta decisión se puede reducir a lo mostrado en la Ecuación (6.1).

$$k^* = \underset{k}{\operatorname{argmax}}(m_{k,i}) \quad (6.1)$$

donde k^* es el usuario elegido para tomar el recurso i , y $m_{k,i}$ es el valor de la métrica del usuario k para el recurso i .

Basado en la métrica utilizada en otras investigaciones [52] [53] se toma la métrica como: la capacidad del canal (elevada a un parámetro α) dividido una expresión relacionada al throughput del usuario y los recursos asignados al momento (elevado a un parámetro β). De esta forma la métrica de un usuario con una capacidad del canal más alta se ve beneficiada, mientras que cuanto más información pase por ese canal la métrica es más perjudicada. A su vez por medio de los exponentes α y β (que pueden variar entre 0 y 1) se puede regular el peso

Capítulo 6. Proportional Fair Generalizado

del numerador y el denominador [54]. En la Ecuación (6.2) se observa el cálculo de la métrica.

$$m_{k,i} = \frac{d_{k,i}(f)^\alpha}{((t_c - 1)T_k(f) + \sum_{j=1}^M \rho_{k,j} \cdot d_{i,j}(f))^\beta} \quad (6.2)$$

donde:

- $m_{k,i}$ es la métrica del usuario k en el RB i .
- $d_{k,j}(f)$ es la tasa alcanzable o la capacidad del canal del usuario k en el recurso i en el subframe f .
- $T_k(f)$ representa el throughput promedio del usuario k hasta el subframe f .
- t_c es la cantidad de slots que se utilizan en la ventana de la media móvil para calcular el valor del throughput promedio.
- $\rho_{k,j}$ es la variable que indica con un “1” si el usuario k tiene asignado el recurso j y “0” en caso contrario.

En la Ecuación (6.2) se puede interpretar el numerador como la recompensa asociada a obtener ese recurso y el denominador como la recompensa ya obtenida tomando en cuenta una ventana de tiempo hacia atrás. Cabe destacar que el denominador tiene en cuenta tanto las decisiones de los slot pasados como las decisiones ya tomadas para el slot actual. Las decisiones son tomadas de forma iterativa sobre cada recurso i y se debe actualizar las métricas que se vean modificadas luego de cada asignación. En la implementación propuesta, por simplicidad, se decide iterar del primer RB en frecuencia al último (existen otras formas válidas de iterar sobre los recursos [53] que poseen mayor complejidad ya que tienen en cuenta todos los recursos para cada decisión).

Es importante definir como se calcula el valor de throughput promedio del usuario k hasta el subframe f ($T_k(f)$). Se calcula como una media móvil de los bits transmitidos en los últimos t_c slots (de todas formas es válido el uso de otras formas de calcular este parámetro [52]). La misma se puede expresar por medio de la Ecuación (6.3):

$$T_k(i+1) = \frac{\sum_{j=1}^{t_c} R_k(i+1-j)}{t_c} \quad (6.3)$$

En la Ecuación (6.3), $R_k(l)$ es la cantidad de bits a transmitir en los recursos entregados en el slot l . La tasa de bits alcanzable es calculada a partir del CQI de la misma forma que es calculada para el modelo de tráfico.

Este algoritmo permite modificar varios parámetros para adecuar el scheduler a los objetivos que se elijan. Por ejemplo modificando los parámetros α y β se puede configurar que tan justo u óptimo (en cuanto a throughput) se prefiere el scheduler. Se puede identificar distintos casos básicos que dan como resultado los distintos extremos del algoritmo:

6.2. Implementación en simulador Vienna 5G SLS

1. $\alpha = 0$ y $\beta = 1$
2. $\alpha = 1$ y $\beta = 0$
3. $\alpha = 0$ y $\beta = 0$

En el primer caso todos los usuarios transmiten la misma cantidad de bits dado que la métrica es inversamente proporcional a la cantidad de bits transmitidos. De esta forma se alcanza una justicia máxima pero al no tomarse en cuenta la tasa alcanzable de los usuarios para la decisión es una asignación cuyo throughput no resulta máximo (dentro de las posibilidades del algoritmo). Si en lugar de tomar como referencia los bits transmitidos, se toma la cantidad de recursos transmitidos entonces el algoritmo sería muy similar a un Round Robin ya que tendería a asignar un recurso a cada uno. En el segundo caso se toma en cuenta únicamente la tasa alcanzable por lo que el algoritmo asignará cada recurso al usuario que sea capaz de sacar más provecho del mismo. En este caso el algoritmo es igual a un Max rate. De esta forma se maximiza el throughput de la red pero disminuye la justicia del algoritmo ya que los usuarios con el CQI más alto son los que pueden llegar a tasas más altas (a su vez son los que tienen mayor capacidad como fue visto en la Sección 3.3). En el último caso donde $\alpha = 0$ y $\beta = 0$ la métrica es igual para todos los usuarios en todos los recursos por lo que el scheduler se comporta como un Round Robin asignando un recurso a cada usuario siguiendo la cola.

El otro parámetro que nos permite ajustar el comportamiento del scheduler es el tamaño de la ventana de la media móvil utilizada para el throughput a largo plazo. La misma cambia la influencia del pasado en el sistema y por lo tanto el dinamismo del mismo.

6.2. Implementación en simulador Vienna 5G SLS

El algoritmo propuesto fue implementado en el simulador Vienna 5G SLS e incluido dentro de las opciones de schedulers disponibles. En la Figura 6.1 se muestra un diagrama de flujo del algoritmo de scheduling implementado. A continuación se explican todos los bloques que se encuentran en dicho diagrama.

Para realizar el scheduling en primer lugar se extrae el CQI del feedback para cada uno de los usuarios y se calcula para todos ellos la media móvil del throughput (el valor por defecto de ventana fue establecido en 20 slots) y un valor inicial del denominador de la métrica (sin elevar al parámetro β). Luego, para cada recurso, se calcula la tasa alcanzable de todos los usuarios en el recurso actual por medio del CQI y a partir del mismo y del denominador calculado previamente se obtiene la métrica. Cuando se tienen los valores de la métrica de todos los usuarios, se buscan los usuarios cuya métrica tenga el valor máximo y en caso de ser varios se toma el usuario que esté más adelante en la cola. Luego se asigna ese recurso en la grilla al ganador, se actualiza la cola pasando dicho usuario al final de la misma y se modifica el valor del denominador de dicho recurso sumándole la tasa alcanzable en el recurso recién asignado. La potencia entregada a cada recurso es asignada

Capítulo 6. Proportional Fair Generalizado

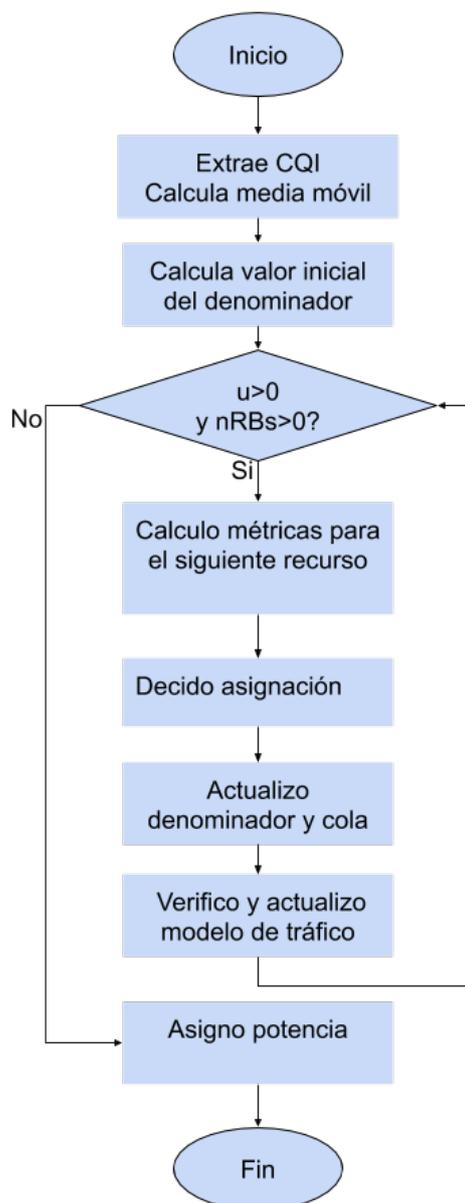


Figura 6.1: Diagrama de flujo del scheduler Proportional Fair Generalizado, la variable u corresponde a la cantidad de usuarios que tienen bits pendientes para transmitir y $nRBs$ el número de recursos libres.

de forma equitativa, se divide la potencia total de la antena sobre la cantidad de recursos y dicho valor es asignado a cada recurso.

Además para este scheduler se tuvo en cuenta el modelo de tráfico por lo que es capaz de asignar limitando el tamaño de los paquetes de acuerdo a lo especificado por cada usuario. Para esto cada usuario tiene un valor de bits a transmitir que, previo a realizar la asignación de los recursos, se carga con el valor en el modelo

6.2. Implementación en simulador Vienna 5G SLS

de tráfico para ese usuario en ese slot y luego por cada recurso que se asigna se lo hace decrecer con el valor estimado de bits a enviar en dicho recurso. Cuando se satisfacen los requisitos de un usuario no se le asignan más recursos.

Cabe destacar que no se toman en cuenta los Resource Allocation Types [55] y que se hace una asignación de PRBs uno a uno. En caso de usar “type0”, se deberían agrupar los PRBs en grupos y por tanto hay menos granularidad de scheduling. Si se usa “type1” se tiene restricción de asignar PRBs en forma contigua y se debe implementar algún algoritmo que asigne en forma randómica esta ubicación contigua. Si siempre se asigna al principio de la banda, por ejemplo, se impone un bias en la interferencia inter-celda.

6.3. Validación del algoritmo implementado

Para la validación del algoritmo Proportional Fair implementado se buscó que fuera sencillo visualizar la dinámica del scheduler, por lo que se realizó una simulación con un chunk de 25 slots y 30 usuarios distribuidos en 7 radiobases. En la Tabla 6.1 se puede observar un resumen de los parámetros utilizado en el escenario. Para asegurar que todos los usuarios requieran recursos en todos los slots y la limitación de tráfico no afecte a la validación, se utilizó un modelo de tráfico constante e igual para cada usuario, donde cada uno pide una cantidad elevada de bits de tal forma de que los usuarios nunca sean satisfechos. De esta manera se controla y se elimina la posibilidad de que un usuario tenga buena métrica pero no requiera ser atendido.

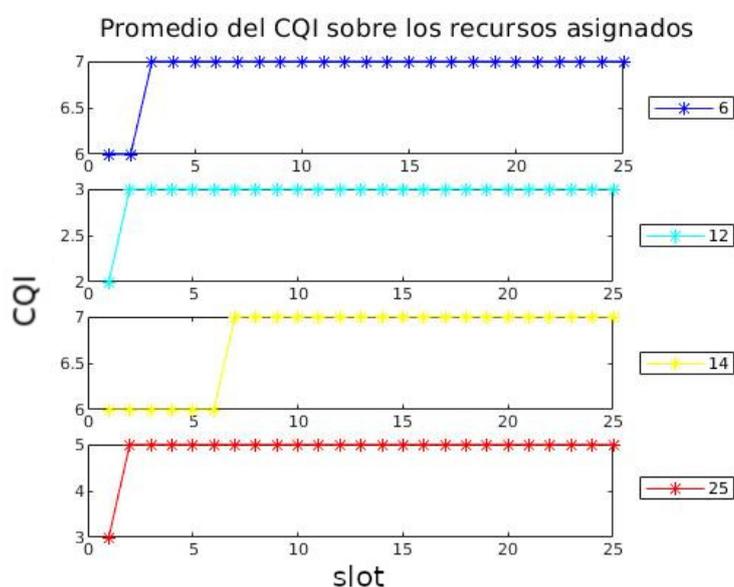


Figura 6.2: Gráfico que representa el CQI por usuario en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$.

Se utiliza en primer lugar un Proportional Fair con $\alpha = 1$ y $\beta = 1$, además la media móvil tiene un tamaño de ventana de 20 slots. En la simulación de ejemplo de se tomó de referencia la radio base número 2 que tenía registrados cuatro usuarios, para los cuales se eligió graficar cuatro magnitudes que se consideran de interés e ilustrativas del algoritmo:

- El valor promedio del CQI de los RBs asignados a cada uno de los usuarios en cada slot (Figura 6.2).
- El valor de la métrica calculada para cada usuario al comienzo de cada slot. Si bien el valor de la métrica se modifica con cada recurso que se le asigna al usuario dentro del slot, tener una idea de la métrica al principio da la información del usuario que llega mejor posicionado al reparto (Figura 6.3).

6.3. Validación del algoritmo implementado

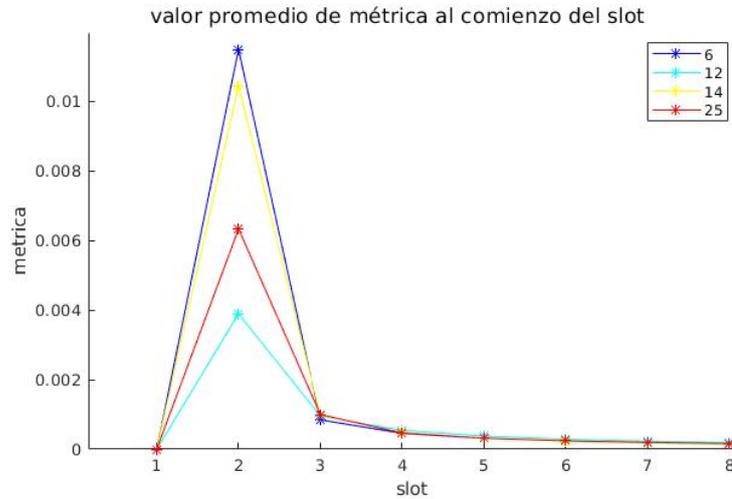


Figura 6.3: Gráfico que representa la métrica por usuario (al comienzo de cada slot) en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$.

- La cantidad de RBs entregados a cada usuario en cada slot (Figura 6.4).
- Los bits a transmitir en los recursos asignados (Figura 6.5).

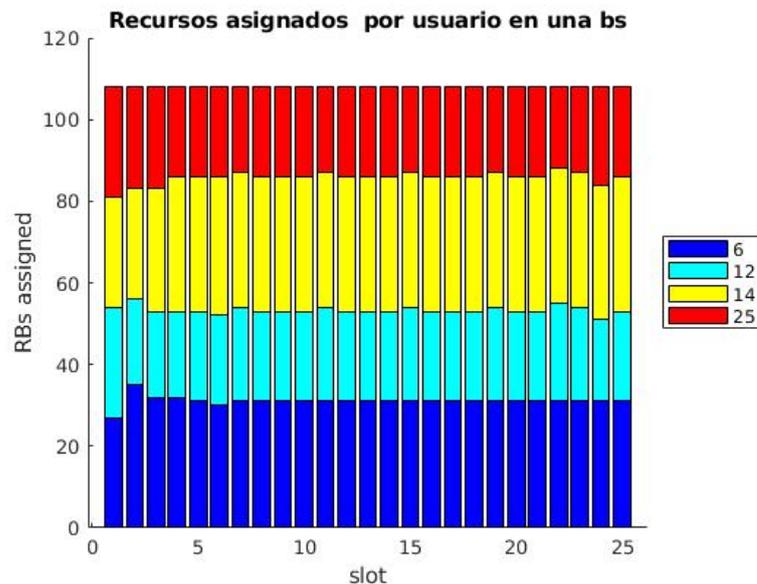


Figura 6.4: Gráfico que representa la asignación de recursos por usuario en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$.

Si se analizan los primeros 5 slots: el primer slot no tiene información del pasado (feedback) por lo que la métrica se toma como 0 para todos los usuarios

Capítulo 6. Proportional Fair Generalizado

Tabla 6.1: Tabla resumen de simulación para validación de Proportional Fair implementado en simulador Vienna 5G SLS

Espacio			
Dimensiones	x=1100	y=1100	z=100
Obstáculos	Espacio abierto		
Tiempo			
slots	25		
feedbackDelay	1		
Transmisión			
Ancho de banda útil	98 %		
Overhead	14 %		
Ancho de banda de portadora	20MHz		
Duración del slot	1 ms		
Scheduler	Proportional Fair		
Frecuencia central	2 GHz		
Radiobases			
Tipo	Macro		
Cantidad	1		
Antena	Omnidireccional		
Usuarios			
Distribución espacial	Poisson		
Cantidad de usuarios estáticos	30		
Trafico de usuarios estáticos	Constante 400000 bits por slot		

y los recursos se asignan de forma igualitaria. En el segundo slot, si se observa en la Figura 6.3 se aprecia que los usuarios con mayor promedio de su métrica al comienzo del slot son: el 6, 14 y el 25. Esto se debe a que poseen un CQI superior al resto de los usuarios como se muestra en la Figura 6.2. Dado que son los que tienen mayor métrica, los usuarios anteriormente nombrados son los más beneficiados en la asignación de los recursos como se observa en la Figura 6.4. También es posible verificar mediante la Figura 6.5 que dada la elevada cantidad de bits asignados, su métrica decae en el tercer slot, lo que muestra que el algoritmo actúa con justicia teniendo en cuenta los recursos asignados en el pasado. A partir del tercer slot la métricas tenderán a ser iguales dado que cuando un usuario tiene mayor métrica se le asignan más recursos provocando un aumento del denominador de la métrica y por lo tanto disminuyendo su prioridad posterior.

Otra validación que permite ver la influencia del numerador y el denominador de la métrica en el algoritmo y el comportamiento de la cola, es estudiar distintos valores de α y β que den comportamientos diferentes aislando diferentes partes del

6.3. Validación del algoritmo implementado

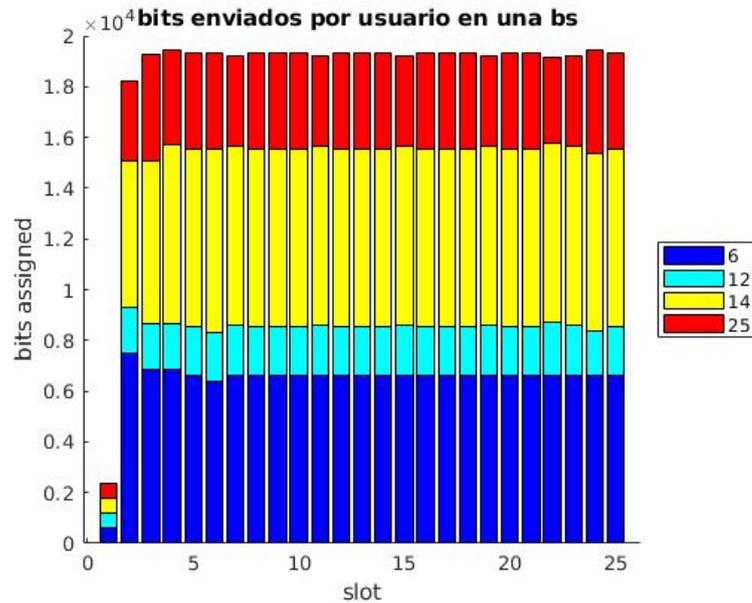


Figura 6.5: Gráfico que representa los bits alojados en los recursos por usuario en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$.

algoritmo. Dichos casos se comentaron en la Sección 6.1. A continuación se detalla el comportamiento observado en el simulador.

Se analiza primero el caso con $\alpha = 0$ y $\beta = 1$. El comportamiento esperado es que sea un algoritmo totalmente justo en cuanto a bits. Se hizo una simulación con dichos valores de peso, y los mismos parámetros especificados en la Tabla 6.1. Se seleccionó para analizar la radio base número 5 en la que se encuentran 6 usuarios recibiendo datos. En la Figura 6.6 se observan los bits transmitidos en los recursos asignados a cada usuario, a simple vista se puede observar dicho comportamiento pero para medirlo se elige utilizar de todos los criterios de justicia mencionados previamente en la Sección 3.3: el índice de justicia de Jain. Dicho valor tomando en cuenta todos los slots de la simulación fue máximo (1) por lo que se puede considerar el algoritmo como justo, lo cual comprueba los supuestos. En otras radiobases se puede observar un comportamiento similar, con valores de justicia iguales a 1 o muy cercanos. En la Figura 6.7 se puede observar que la asignación no es equitativa en cuanto a recursos asignados, esto se debe a que se están tomando en cuenta los bits transmitidos y no la cantidad de recursos.

En cambio si se ajustan los valores a $\alpha = 1$ y $\beta = 0$ (como se observa en la Figura 6.8) en el slot 1 donde el scheduler no tiene información de capacidad del canal asigna de manera equitativa, pero luego solo algunos usuarios reciben recursos. Esto se debe a que el scheduler asigna cada recurso al usuario que mayor tasa de bits logre en caso de utilizarlo. Si se analizan los valores de CQI en la Figura 6.9 se puede observar que los usuarios 28, 15 y 2 son los que poseen mayor promedio de CQI lo que se traduce a una capacidad de canal mayor. El índice de

Capítulo 6. Proportional Fair Generalizado

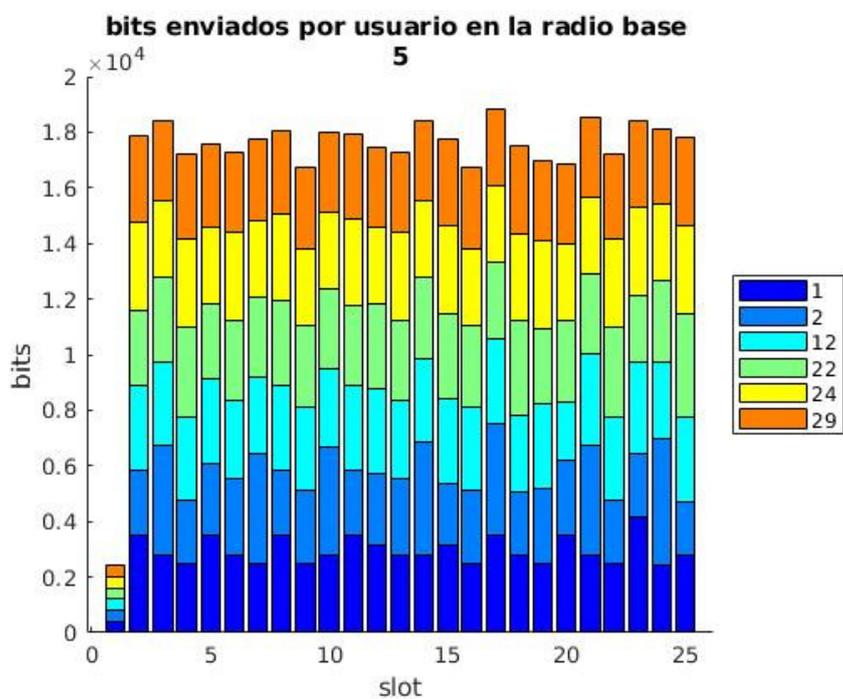


Figura 6.6: Gráfico de barras de bits transmitidos por usuario en radiobase 5 en simulación de algoritmo Proportional Fair con $\alpha = 0$ y $\beta = 1$.

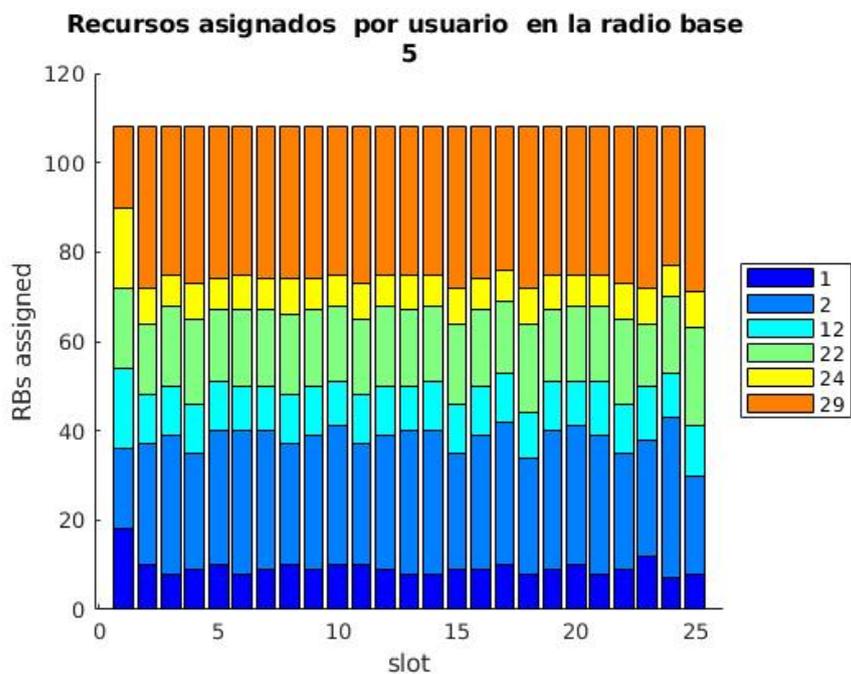


Figura 6.7: Gráfico de barras de recursos asignados por usuario en radiobase 5 en simulación de algoritmo Proportional Fair con $\alpha = 0$ y $\beta = 1$.

6.3. Validación del algoritmo implementado

Jain fue de 0,1410 un valor muy inferior al conseguido con los otros valores de α y β . Esto es acorde con lo esperado dado que no se tiene en cuenta la cantidad de bits transmitida sino que se tiene en cuenta únicamente la capacidad del canal en cada recurso. Por lo que se verifica que el denominador da la justicia al algoritmo y el numerador la eficiencia de los recursos.

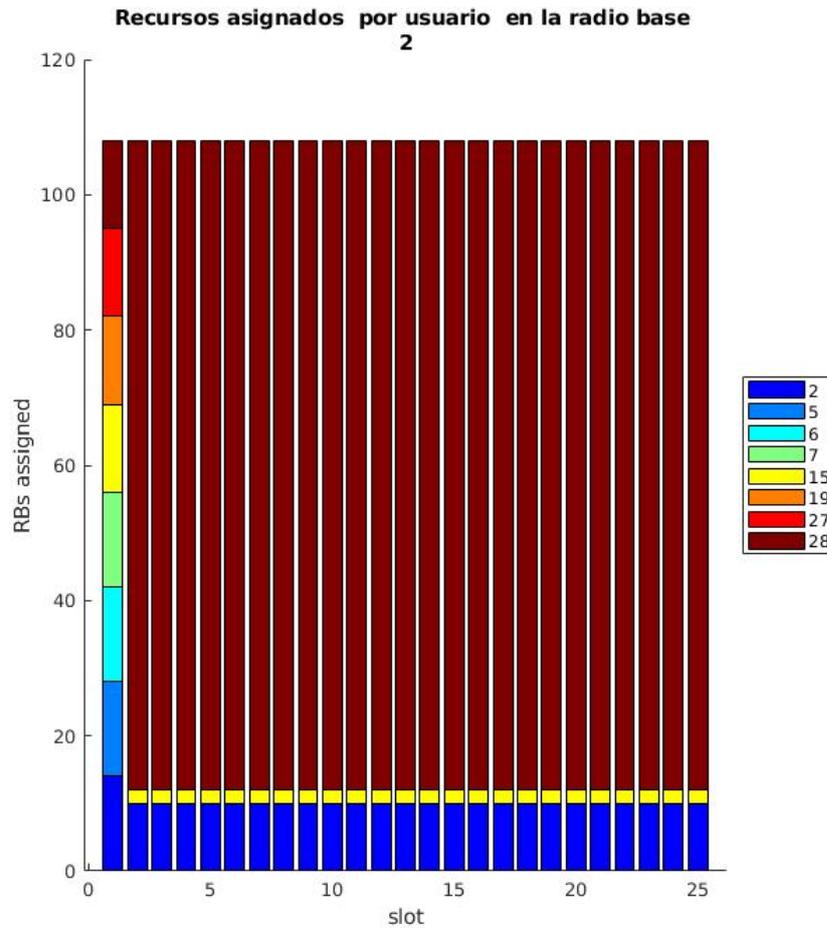


Figura 6.8: Gráfico de barras de recursos asignados por usuario en radiobase 2 en simulación de algoritmo Proportional Fair con $\alpha = 1$ y $\beta = 0$

El último caso a estudiar es la métrica igual para todos y constante, colocando $\alpha = 0$ y $\beta = 0$ (métrica 1 en todos los recursos). Con este caso se verifica el funcionamiento del empate y el manejo de la cola de usuarios. En la Figura 6.10 se puede observar que la asignación de recursos a los usuarios en la BS se reparten de forma equitativa sobre todos los usuarios. Esto es acorde con lo esperado dado que todos los usuarios tienen la misma métrica y se le asigna un recurso a cada uno, enviando al final de la cola al usuario que se le asigna un recurso. Por lo que se valida el funcionamiento de la cola y el desempate de métricas.

Capítulo 6. Proportional Fair Generalizado

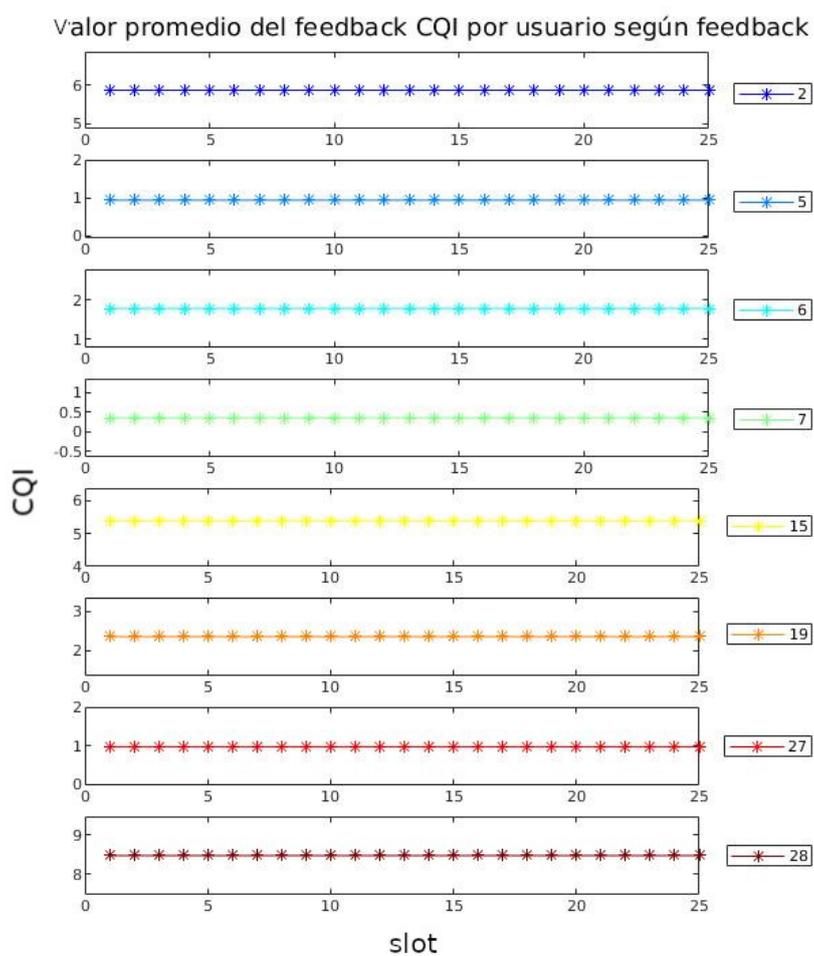


Figura 6.9: Gráfico del CQI promedio por usuario extraído del feedback en la radiobase 5 para la simulación con algoritmo Proportional Fair de $\alpha = 1$ y $\beta = 0$.

6.3. Validación del algoritmo implementado

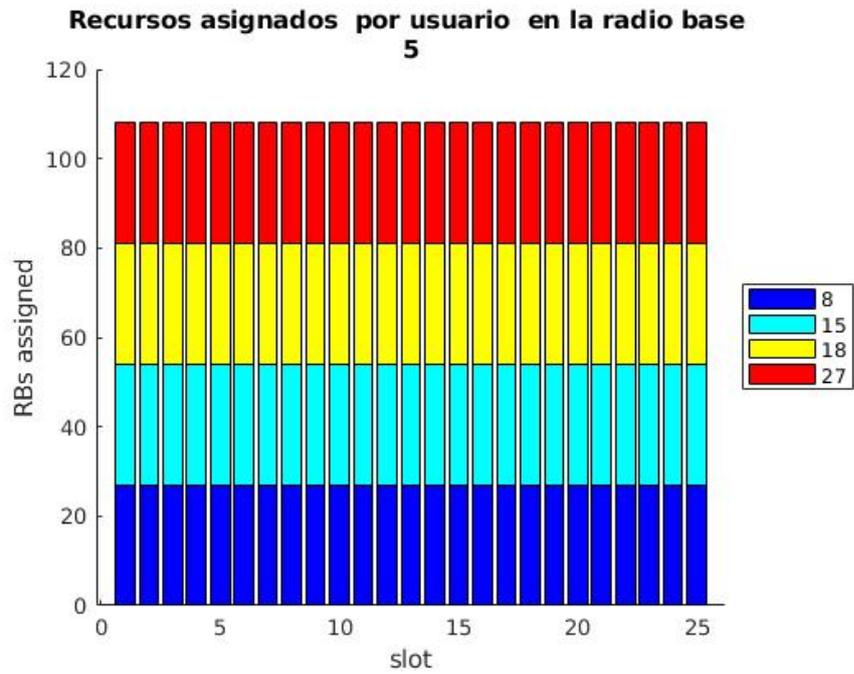


Figura 6.10: Gráfico de barras de recursos asignados por usuario en radiobase 5 en simulación de algoritmo Proportional Fair con $\alpha = 0$ y $\beta = 0$

6.4. Resumen del Capítulo

En el presente capítulo se presentó la variante del algoritmo Proportional Fair implementado, los objetivos del mismo, las características principales y las particularidades del diseño. Además se comentaron detalles de la implementación sobre el simulador Vienna 5G SLS y una validación de la misma. En los capítulos siguientes se detallarán implementaciones cuyo funcionamiento está por completo basado en las nuevas características de la tecnología 5G NR utilizadas para brindar servicios diferenciales.

Capítulo 7

Grilla con minislots y Proportional Fair con puncturing

Para cumplir con el objetivo de la implementación del algoritmo Proportional Fair con Puncturing (mecanismo que prioriza usuarios URLLC por sobre eMBB) fue necesario adaptar la grilla de recursos del simulador para que funcione con minislots (como se mencionó anteriormente el simulador Vienna 5G SLS no contaba con la funcionalidad de minislots o sTTI). En la siguiente sección se explica como se realizó dicha implementación y posteriormente se muestra la validación. En el presente capítulo también se aborda la implementación y validación del algoritmo Proportional Fair con Puncturing.

7.1. Grilla con minislots

Una carencia que presenta el simulador Vienna 5G SLS es que no implementa múltiples espaciados entre subportadoras (concepto que se explica en el Capítulo 2). La implementación que hace el simulador se basa en subportadoras espaciadas 15 kHz (buscando compatibilidad con LTE). Para trabajar con otras numerologías se deben crear diferentes ResourceGrid (en el directorio +resourceGrid) y cambiar los parámetros que caracterizan cada una. En este proyecto no se agregaron más numerologías por considerar que se apartaba del alcance. Lo que se realizó fue adaptar la grilla a los requerimientos de NR funcionando con espaciado entre subportadoras de 15 kHz.

7.1.1. Implementación grilla NR y grilla con minislots

Como se mencionó en la Sección 4.1.1 dentro del paquete “+parameters” se encuentra el subpaquete “+resourceGrid” donde se definen los diferentes tipos de grillas de recursos que puede manejar el simulador (“LTE.m”, “NR_minislot.m” y “NR.m”). El campo más representativo de la grilla es el “userAllocation” que es la matriz que contiene los IDs de los usuarios a los que se asignan los recursos (los demás parámetros que conforman el objeto grilla se encuentran descritos en el

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

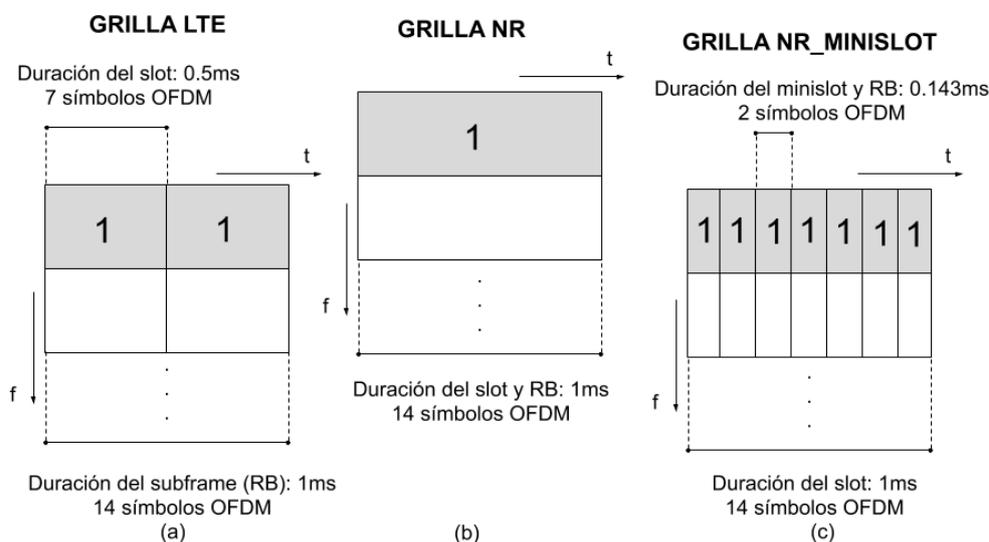


Figura 7.1: Representación de las grillas de recursos que maneja el simulador. (a) Grilla LTE, (b) Grilla NR, (c) Grilla NR con minislots.

final de la Sección 4.1.3). En este documento se refiere a la matriz “userAllocation” como grilla (haciendo abuso de notación para simplificar la lectura). La grilla es una matriz de tantas filas como RB en frecuencia existan y tantas columnas como RB se ubiquen en el tiempo (por subframe en LTE o slot en NR).

La grilla está configurada, por defecto, según el script “LTE.m” que define la duración del subframe (en el script se lo menciona como slot) en 1 ms (medida de tiempo en la que el scheduler asigna los recursos) y 14 símbolos OFDM. La duración del slot (en el script aparece como RB) en 0,5 ms con 7 símbolos OFDM. A modo de ejemplo: al asignar un RB al usuario con ID 1 en la grilla aparece el número 1 (ID) en ambos slots del RB (en LTE: 1 RB = 1 subframe = 2 slots) como se muestra en la Figura 7.1 (a).

La grilla de recursos definida en “LTE.m” no es compatible con lo estipulado para NR por lo que se creó el script “NR.m”. En el mismo se define en 1 ms tanto la duración de slot (medida de tiempo en la que el scheduler asigna los recursos) como la duración del RB en el tiempo. Se fija en 14 la cantidad de símbolos OFDM por RB y de esta forma se obtiene la grilla según los requerimientos de NR [4]. En la Figura 7.1 (b) se ve como quedaría representado el caso de que se le entregue un recurso al usuario con ID (IDentification) 1.

En el script “NR_minislot.m” la nueva grilla implementada corresponde a un RB en frecuencia y un minislot de 2 símbolos en el tiempo. Según la definición que se manejó en la Sección 2.2.6 la longitud (en símbolos OFDM) de los minislots se puede elegir entre 2 y 13. Entre las opciones se priorizó las cantidades de símbolos que sean divisoras de 14 para poder tener una grilla regular (que todos los slots tengan la misma cantidad de minislots) para que la implementación de la grilla no agregara mayor complejidad al scheduler. Se eligió que el minislot tenga longitud de 2 símbolos para hacer notoria la funcionalidad de sTTI (la opción de 2 minislots

7.1. Grilla con minislots

Tabla 7.1: Tabla resumen del escenario para la validación de las grillas de recursos

Espacio			
Dimensiones	x=500	y=500	z=100
Obstáculos	Espacio abierto		
Tiempo			
Slots	25		
FeedbackDelay	1		
Transmisión			
Eficiencia espectral	98 %		
Overhead	14 %		
Ancho de banda de portadora	20MHz		
Duración del slot	1 ms		
Scheduler	Proportional Fair		
Radiobases			
Tipo	Macro		
Cantidad	1		
Antena	Omnidireccional		
Usuarios			
Distribución espacial	Poisson		
Cantidad de usuarios en movimiento	1		
Tráfico de usuarios en movimiento	Constante 12000 bits por slot		
Velocidad	50 km/h		
Cantidad usuarios estáticos	5		
Trafico de usuarios estáticos	Constante 13000 bits por slot		

de 7 símbolos resulta en una matriz análoga a la que implementa la grilla LTE del simulador). En la Figura 7.1 (c) se ve como quedaría representado el caso de que se le entregue un recurso a un usuario eMBB con ID 1. Como se trata de un usuario de servicio banda ancha (eMBB) se le entregan los 7 recursos (pertenecientes a los 7 minislots) completando el equivalente a un RB de la grilla NR. La implementación de esta grilla con minislots permite la incorporación de usuarios URLLC. A dichos usuarios se le asignan los RB (pertenecientes a los minislots) que solicitan y en caso de existir un usuario eMBB en dicho RB se le pisan los datos (como se explicó en la Sección 2.2.6). En las simulaciones que se realizaron la grilla tiene dimensión 106x7 donde hay 106 RB en frecuencia y 7 minislots en el tiempo.

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

obj.rbGrid.DL.userAllocation			obj.rbGrid.DL.userAllocation			obj.rbGrid.DL.userAllocation			obj.rbGrid.DL.userAllocation		
	1	2		1	2		1	2		1	2
1	-1		1	-1		1	-1		98	2	
2	-1		2	-1		2	2		99	3	
3	-1		3	-1		3	-1		100	4	
4	-1		4	-1		4	-1		101	5	
5	-1		5	-1		5	-1		102	6	
6	-1		6	-1		6	-1		103	1	
7	-1		7	-1		7	-1		104	2	
8	-1		8	-1		8	-1		105	3	
9	-1		9	-1		9	-1		106	4	
10	-1		10	-1		10	-1		107	5	
11	-1		11	-1		11	-1		108	6	
12	-1		12	-1		12	-1				

Grilla inicializada Asignación del primer recurso Asignación del segundo recurso ... Asignación del último recurso

Figura 7.2: Evolución de la grilla de recursos correspondiente al primer slot de la simulación con grilla NR.

7.1.2. Validación grilla NR y NR_minislot

Para validar las grillas NR y NR_minislot se comparan los resultados obtenidos a partir de utilizar ambas grillas en simulaciones. La idea de la validación es utilizar el mismo escenario en las dos simulaciones (una con cada grilla) y estudiar si los resultados son coherentes y afirmativamente análogos entre sí. El escenario cuenta con una BS, 5 usuarios estáticos y un usuario en movimiento (todos los usuarios son de tipo eMBB). En cada simulación se realizan breakpoints para verificar la forma en la que se rellena la grilla de recursos. En la Tabla 7.1 se describe el escenario a utilizar en la validación de las grillas de recursos.

Validación grilla NR

Se corre la simulación fijando los parámetros mostrados en la Tabla 7.1 en el escenario NR como iniciador y se colocaron dos breakpoints. De esta forma, con el primer breakpoint, se puede ver slot a slot como se va llenando la grilla con los ID de los usuarios a los que se asignan recursos. Y con el segundo breakpoint se obtiene la grilla completa luego de finalizar el slot. A modo de ejemplo, en la Figura 7.2, se observa la forma en que se va completando la grilla de recursos correspondiente al primer slot de la simulación. En un primer momento la grilla está inicializada con -1 en todos los recursos, indicando que dicho recurso no se encuentra asignado. Posteriormente la asignación de recursos prosigue de RB en RB. En la Figura 7.2 se verifica el funcionamiento explicado en la sección anterior e ilustrado en la Figura 7.1 (b).

En la Figura 7.3 se visualiza la distribución espacial de la BS y los usuarios que participan en la simulación.

Validación grilla NR_minislot

Para validar la grilla NR_minislot se configura el escenario NR_minislot según las características presentadas en la Tabla 7.1. Una vez realizada la simulación se obtuvo la distribución de usuarios y la BS como se muestra en la Figura 7.4.

7.1. Grilla con minislots

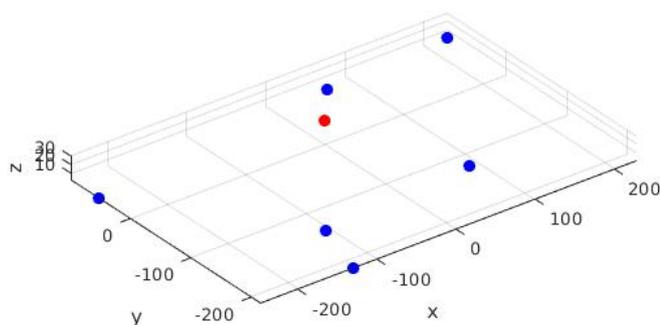


Figura 7.3: Distribución espacial de BS y usuarios en la simulación con grilla NR.

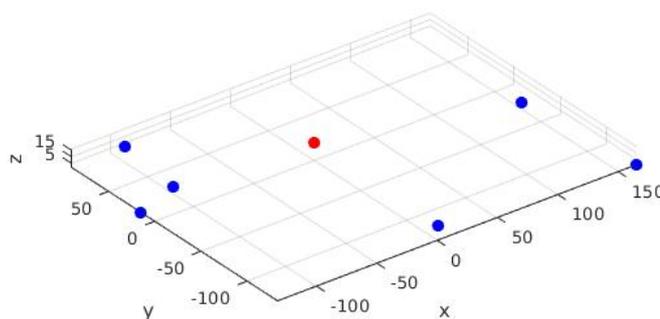


Figura 7.4: Distribución espacial de BS y usuarios en la simulación con grilla NR_minislot.

Se utilizan los mismos breakpoints que los descritos en la sección anterior y se consigue analizar el paso a paso del llenado de la grilla (de RB en RB) lo que se muestra en la Figura 7.5. En la misma se verifica la estructura de matriz con 7 minislots que se explicó en la Subsección 7.1.1 demostrando un funcionamiento correcto y de acuerdo a lo esperado.

Comparación entre resultados de ambas simulaciones

Debido a las pequeñas dimensiones del escenario y por poseer una BS atendiendo solo a 6 usuarios los mismos presentan CQI alto en sus recursos. Esto coincide

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

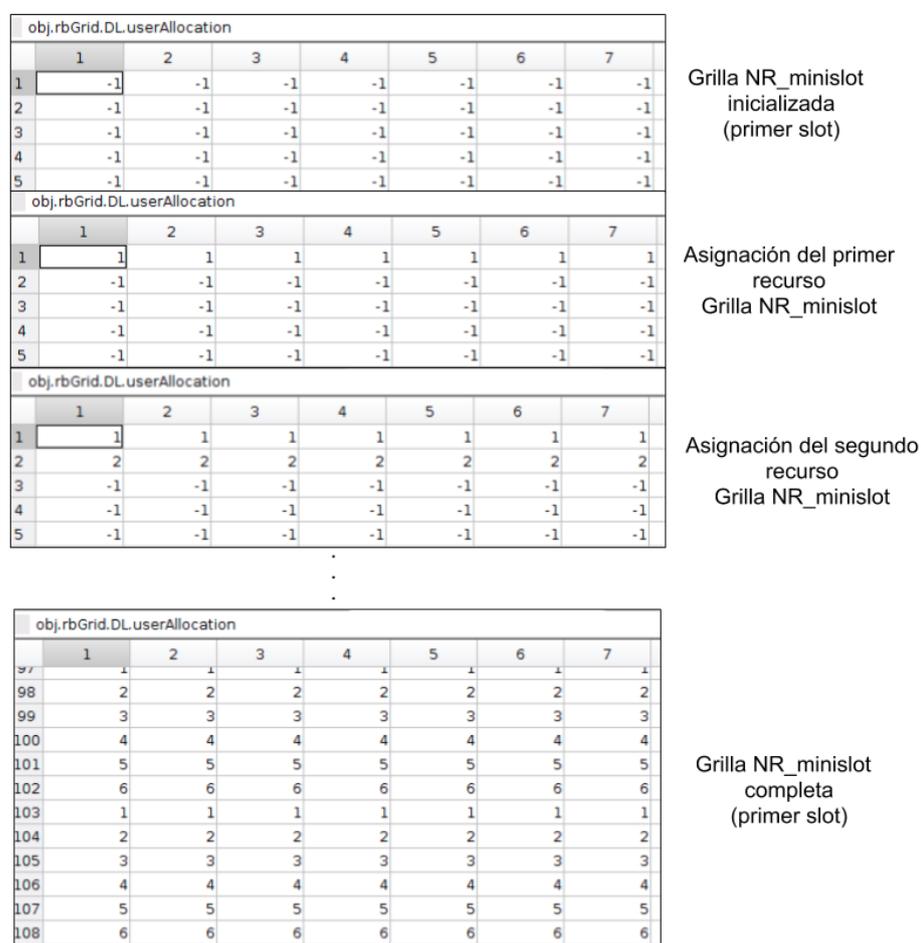


Figura 7.5: Evolución de la grilla de recursos correspondiente al primer slot de la simulación con grilla NR_minislot.

en ambas simulaciones como se visualiza en la Figura 7.6 donde el promedio de CQI en todos los recursos es 15. Como el CQI es exactamente el mismo para todos los usuarios en todos los recursos todos los usuarios tienen la misma métrica (se usó el algoritmo Proportional Fair para asignar los recursos) por esta razón se ve gran paridad en la asignación. En la Figura 7.7 se aprecia que para la simulación con grilla NR en los primeros 9 slots se utilizan todos los recursos (108) y a partir del décimo slot se estabiliza utilizando menos RB (se salda el déficit contraído en los primeros slots). En dicha simulación se utilizaron entre 15 y 19 RB por usuario por slot cubriendo entre 12000 y 15000 bits (como se muestra en la Figura 7.8).

La asignación de recursos en la simulación con grilla NR_minislot es más regular, se asignan 126 recursos (que corresponden a 18 recursos eMBB) a todos los usuarios en todos los recursos (como se ve en la Figura 7.7). Los 126 recursos transportan 14450 bits (por usuario en cada slot), lo que cubre los requerimientos planteados (ver Figura 7.8).

7.1. Grilla con minislots

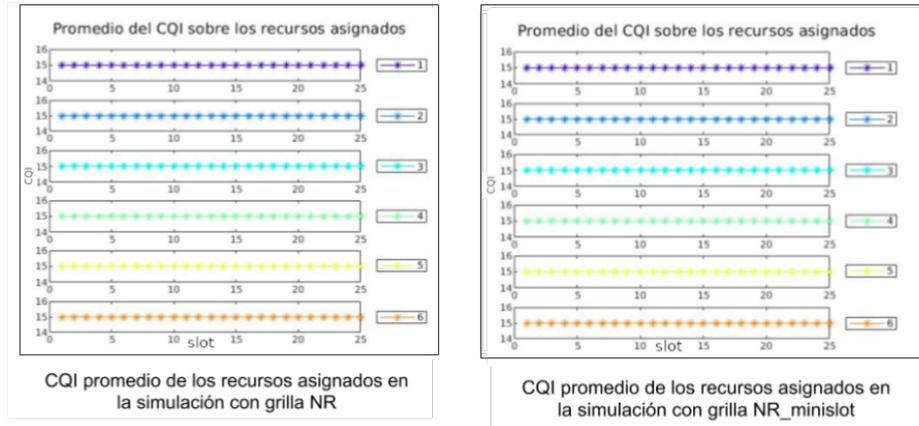


Figura 7.6: CQI promedio de los RB asignados a los usuarios en las simulaciones con grilla NR y NR_minislot.

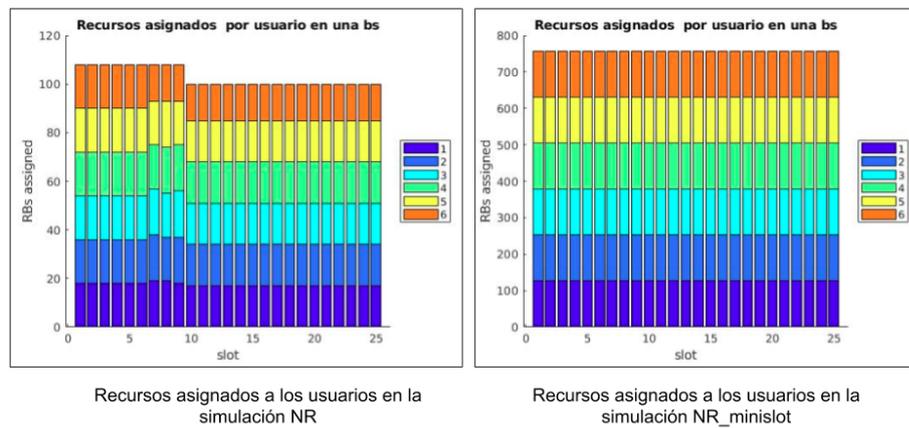


Figura 7.7: RB asignados a los usuarios en las simulaciones con grilla NR y con grilla NR_minislot.

A modo de conclusión se aprecia que la cantidad de RB asignados en ambas simulaciones son entre 15 y 19 para NR y 18 para NR_minislot. Si se divide la cantidad de bits transmitidos entre la cantidad de recursos asignados (dado que el CQI de los usuarios es el mismo) se comprueba que la cantidad de bits transmitidos por recurso es muy cercana entre ambos casos por lo que se valida que el agregado de los minislots no modifica el comportamiento de los usuarios eMBB. También se comprobó que el llenado de la grilla funciona de acuerdo a lo previsto. El funcionamiento de la grilla con minislots funcionando con usuarios URLLC se valida junto con el algoritmo Proportional Fair con Puncturing en la Sección 7.2 por ser funcionalidades dependientes una de otra.

A continuación se detalla la implementación y validación del algoritmo Proportional Fair con Puncturing. También se comentan las diferentes funcionalidades que fue necesario agregar al simulador para que el algoritmo actúe de acuerdo a lo

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

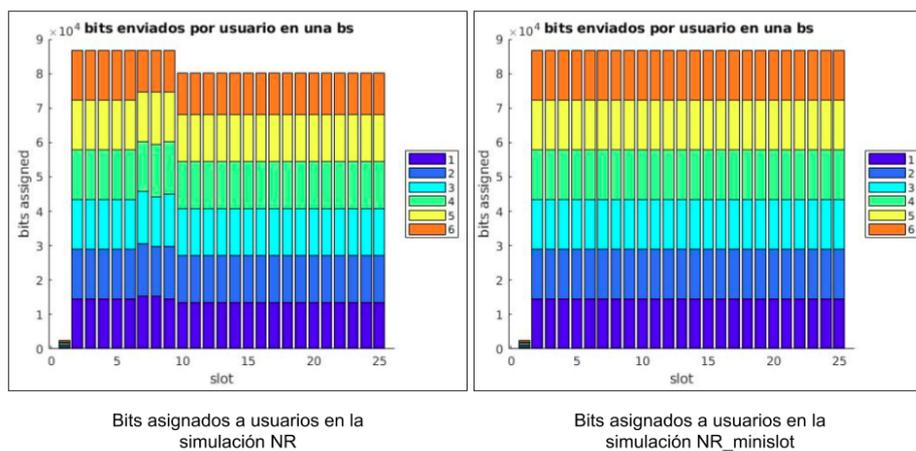


Figura 7.8: Bits asignados a los usuarios en las simulaciones con grilla NR y con grilla NR_minislot.

esperado.

7.2. Proportional Fair con Puncturing

El segundo algoritmo de scheduler implementado en el simulador fue un Proportional Fair con Puncturing (véase Sección 2.2.6 donde se introduce el concepto de puncturing). Como se mencionó anteriormente el simulador no contaba en su estructura con la posibilidad de introducir un estudio a nivel de minislot por lo que se realizaron las modificaciones desarrolladas en la Subsección 7.1.1.

Cabe aclarar que el punctured scheduling es un posible uso de los minislots, pero que estos fueron previstos por la norma como herramienta para distintos posibles usos.

7.2.1. Agregado de usuarios URLLC

Para realizar simulaciones con usuarios URLLC (servicios de baja latencia) lo primero que se hizo fue encontrar una forma de identificarlos. Para esto se agregó un campo llamado “UserType” al objeto “user” (véase Sección 4.1.2). Este campo es un valor entero el cual toma el valor 1 para usuarios del tipo eMBB y 2 para usuarios URLLC. El mismo se agregó en la inicialización de los usuarios en el script “SimulationSetup” dentro del paquete “+simulation” en la función “createUsers” (en esta función se asigna el UserType, junto con el ID, especificado para ese usuario). También fue necesario agregar el campo UserType en el escenario utilizado para la simulación (en la sección de “User”). Otro aspecto que caracteriza a los usuarios URLLC es el modelo de tráfico. Dicho modelo de tráfico tiene una tercera dimensión que corresponde a los minislots contenidos en cada slot (como se comentó en la Sección 5.1). Las coordenadas para acceder a la matriz del modelo de tráfico son (chunk, slot, minislot) y en cada celda se encuentra especificado el requerimiento de tráfico medido en RB (véase Sección 5.1).

7.2.2. Implementación de puncturing en el simulador

La estructura de tiempo del simulador no permite realizar una interrupción de una transmisión dado que la mínima unidad de tiempo de asignación de recursos es el slot. Por lo tanto no puede introducirse la técnica de puncturing asignando minislots sin cambiar dicha estructura básica. Para sortear este problema lo que se hizo fue aprovechar que mediante el traffic model se conocen de antemano los slots en los que los usuarios URLLC reciben datos. En dichos slots se sobrescriben los recursos asignados por el scheduler a usuarios eMBB, pero esto se hace al comienzo del slot (no es posible ejecutarlo en tiempo real en el transcurso del slot). A los efectos de los resultados es indistinto el momento en que ocurre la interrupción ya que el usuario interrumpido (eMBB) pierde la totalidad de los datos transmitidos en ese recurso [8], además no se considera el overhead extra generado por la señalización en el PDCCH de los recursos asignados a usuarios URLLC. Para simular el impacto de la transmisión HARQ se desasignan los recursos asignados al usuario eMBB (esto es posible por tratarse de una simulación). De esta forma es posible realizar un estudio del impacto generado en una red con usuarios URLLC y estudiar diferentes políticas para elegir cual es el recurso a sobrescribir.

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

Para hacer el scheduling se separó el problema en dos subproblemas (como se plantea en el paper “Punctured Scheduling for Critical Low Latency Data on a Shared Channel with Mobile Broadband” [8]). Por un lado la asignación de recursos a los usuarios eMBB sin tomar en cuenta a los usuarios URLLC y por otro lado la asignación de recursos mediante la técnica de puncturing a los usuarios de URLLC. Para la solución del primer problema se consideró que el algoritmo de Proportional Fair es el más adecuado dado que beneficia a los usuarios más afectados por el puncturing (el promedio de recursos asignados baja en dichos usuarios lo que hace aumentar la métrica). Para esta parte los recursos se reparten de forma idéntica a lo expuesto en la Sección 6.1. En cuanto a la asignación de los recursos para usuarios URLLC se realizó basado en lo propuesto en el trabajo de K. I. Pedersen [8] donde se utiliza la siguiente métrica:

$$\frac{r_{v,p}[n]}{R_v[n]} \cdot W_p^\gamma[n] \quad (7.1)$$

Donde:

- $r_{v,p}$ es la tasa alcanzable por el usuario “v” (URLLC) en el RB “p” en el slot “n”.
- $R_v[n]$ es la tasa promedio de transmisión del usuario “v” previa al slot “n”.
- $W_p^\gamma[n]$ es la cantidad de bits a transmitir esperada por el usuario eMBB en el RB “p” y en el slot “n”.

Variando el parámetro γ se pueden lograr diferentes políticas para la asignación de recursos:

- BR (Best Resources) $\gamma = 0$: Si γ es nulo se observa que a un usuario se le asignan los recursos cuya tasa de transmisión se maximice, es decir los recursos que el usuario tenga mayor CQI.
- HeU (Highest eMBB User) $\gamma = 1$: Además de buscar buen CQI en el recurso a asignar se priorizan los recursos cuyo MCS sea mayor para el usuario que lo posee. De esta forma se protegen usuarios que se encuentran en el borde de la celda.
- LeU (Lowest eMBB User) $\gamma = -1$: Además de buscar que el usuario tenga buen CQI en el recurso a asignar, se priorizan los recursos cuyo MCS sea mínimo. Con esta política se busca minimizar el impacto sobre el throughput de la BS tomando los recursos que en la situación actual utilizarían un MCS bajo.

Se implementó el scheduling teniendo en cuenta los tres criterios (cambiando el valor de gamma). En la Subsección 7.2.3 se analizan los tres casos. Dada la falta en el simulador de un cálculo de throughput adecuado a este tipo de usuarios (URLLC), no se tomará en cuenta el denominador de la métrica. En su lugar se tomarán los usuarios uno a uno siguiendo una cola.

7.2. Proportional Fair con Puncturing

Para implementar el algoritmo de scheduler con puncturing se deben seguir dos pasos: en primer lugar se realiza la asignación de recursos sobre los usuarios eMBB (con UserType 1) de forma análoga a lo explicado para el Proportional Fair (veáse Capítulo 6). Al terminar la asignación de usuarios de banda ancha se recorren los minislots verificando los usuarios que interrumpen en cada uno y se les asignan los recursos demandados siguiendo el criterio elegido (según el γ utilizado) como se explicó anteriormente. Cuando un usuario eMBB es interrumpido, la BS tiene dos posibilidades: una es retransmitir el recurso entero y otra es retransmitir solo la parte afectada. En la implementación realizada se toma como criterio la pérdida de todo el recurso por lo que se le retira la asignación del resto del recurso al usuario (eMBB). Posteriormente se reparten los recursos correspondientes a los minislots que requiere el usuario URLLC. En el caso que se utilice como criterio el BR ($\gamma = 0$) o el LeU ($\gamma = -1$) una vez que el usuario URLLC interrumpe a un usuario eMBB el recurso queda perdido por completo (para el usuario eMBB) pero puede utilizarse por usuarios URLLC que requieran recursos en dicho slot en minislots posteriores (se reciclan los minislots). Si se utiliza el criterio HeU ($\gamma = 1$) no se produce el reciclaje de recursos en los slots pisados ya que el criterio HeU se basa en utilizar recursos de los mejores usuarios eMBB (una vez que se pisa un recurso eMBB ya no pertenece a dicho usuario).

En la siguiente sección se desarrolla la validación del algoritmo funcionando con usuarios eMBB y usuarios URLLC. El caso del algoritmo funcionando sólo con usuarios eMBB es análogo al descrito en el Capítulo 6 mientras que el caso del algoritmo funcionando solamente con usuarios URLLC no tiene sentido práctico (la implementación se basa en la interacción de usuarios eMBB y usuarios URLLC).

Una aclaración importante a realizar es que para las implementaciones realizadas en este capítulo se consideran TB de 2 símbolos. No se tienen en cuenta los overheads generados y tampoco la localización de los símbolos dentro del slot.

7.2.3. Validación Proportional Fair con Puncturing

Para las validaciones que se presentan en esta sección se utilizó el archivo “PropFairScheduler.m” con los exponentes del numerador y denominador iguales a la unidad. Se comienza por validar los diferentes casos propuestos para la asignación de recursos de los usuarios URLLC. Se utiliza el mismo escenario en las tres simulaciones variando únicamente el valor del parámetro γ (con valores 0, 1 y -1). El escenario elegido cuenta con 7 BS (representadas en rojo en las imágenes) equidistantes (750 m), 20 usuarios eMBB (representados en azul) y 10 usuarios URLLC (representados en verde). Todos los elementos son estáticos y distribuidos según Poisson. La configuración completa del escenario se describe en la Tabla 7.2. Para visualizar el momento en que se asignan recursos a los usuarios URLLC se colocan dos breakpoints en el archivo “PropFairScheduler.m” en los momentos que se asignan recursos a los usuarios URLLC. En la línea 205 cuando el usuario URLLC no pisa a un usuario eMBB y en la línea 218 cuando sí pisa a un usuario eMBB.

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

Tabla 7.2: Tabla resumen del escenario para la validación de la asignación de recursos a usuarios URLLC

Espacio			
Dimensiones	x=1500	y=1500	z=100
Obstáculos	Espacio abierto		
Tiempo			
Slots	10		
FeedbackDelay	1		
Transmisión			
Eficiencia espectral	98 %		
Overhead	14 %		
Ancho de banda de portadora	20MHz		
Duración del slot	1 ms		
Scheduler	Proportional Fair con Puncturing		
Radiobases			
Tipo	Macro		
Cantidad	7		
Antena	Omnidireccional		
Usuarios			
Distribución espacial	Poisson		
Cantidad de usuarios eMBB	20		
Tráfico de usuarios eMBB	Constante 1300 bits por slot		
Cantidad usuarios URLLC	10		
Trafico de usuarios URLLC	Ráfaga 2 recursos por minislot		

Caso BR (Best Resources)

Una vez ejecutada la simulación se obtuvo la distribución de usuarios que se muestra en la Figura 7.9. Se eligió trabajar con la BS 4 que cuenta con 5 usuarios eMBB (IDs): 7, 9, 11, 13 y 14 y con 2 usuarios URLLC (IDs): 26 y 29. Durante la simulación se analizó la forma en la que se rellenó la grilla de recursos.

En este ejemplo se estudió lo sucedido en el slot número 3. En primer lugar se buscó el tmodel de los usuarios de URLLC (26 y 29) para conocer los requerimientos que tenían. El usuario 26 presentaba requerimiento de 2 RB en los minislots: 2, 5 y 6, mientras que el usuario 29 requería 2 RB en el minislot 1. El requerimiento total de recursos del usuario 26 para el slot 3 es de 6 RB mientras que el usuario 29 requiere 2 RB en dicho slot. En la Figura 7.10 se observa que se cumplió la

7.2. Proportional Fair con Puncturing

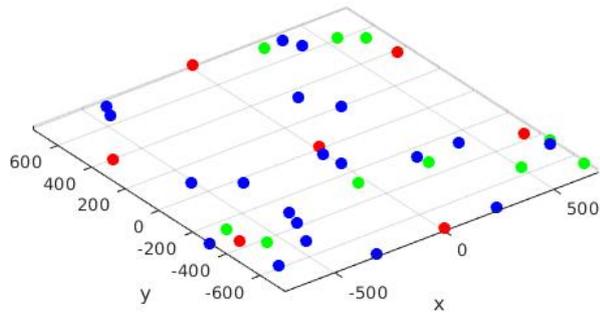


Figura 7.9: Escenario para la validación del Proportional Fair con Puncturing funcionando con BR. Con rojo se representan las BS, con verde los usuarios URLLC y con azul los usuarios eMBB.

asignación de recursos.

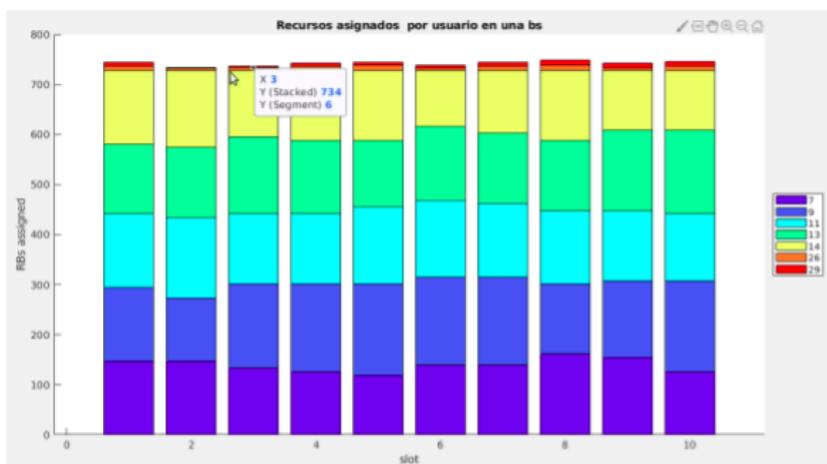
Con la ayuda de los breakpoints fue posible analizar la evolución de la grilla de recursos. En el primer minislots fue el usuario 29 que requirió dos RB, los mismos le fueron concedidos en los recursos 15 y 16 pisando los recursos al usuario 13 (eMBB). Como se explicó anteriormente el usuario 13 pierde los recursos que se le habían asignado. Posteriormente se asignan los recursos correspondientes al usuario 26 en el segundo minislots. En este caso se pisaron los recursos asignados a los usuarios 14 y 7 de los RB 46 y 47 respectivamente. Esto se observa en la Figura 7.11. Finalmente se asignaron los RB correspondientes al usuario 26 en los minislots 5 y 6. Los mismos se asignaron también en los recursos 46 y 47, de esta forma se reutilizan los recursos que se habían perdido (como se muestra en la Figura 7.12).

A partir de la Figura 7.13, donde se visualiza el CQI promedio de los RB asignados a los usuarios, es posible verificar si los recursos elegidos para transmitir la información URLLC es adecuada. En primer lugar se observa que el usuario 13 tiene un CQI promedio de 9 en el slot 3, siendo el más alto (junto con el 11). El usuario 29 (que pisó los recursos 15 y 16) del usuario 13 obtuvo un CQI promedio de 15. Por su parte, el usuario 26, le pisó el recurso 46 al usuario 14 y el 47 al 7 obteniendo un CQI promedio de 8 en sus recursos. El promedio del CQI del usuario 26 es similar al del usuario 7 (CQI=7) y al del usuario 14 (CQI=8) en el slot 3.

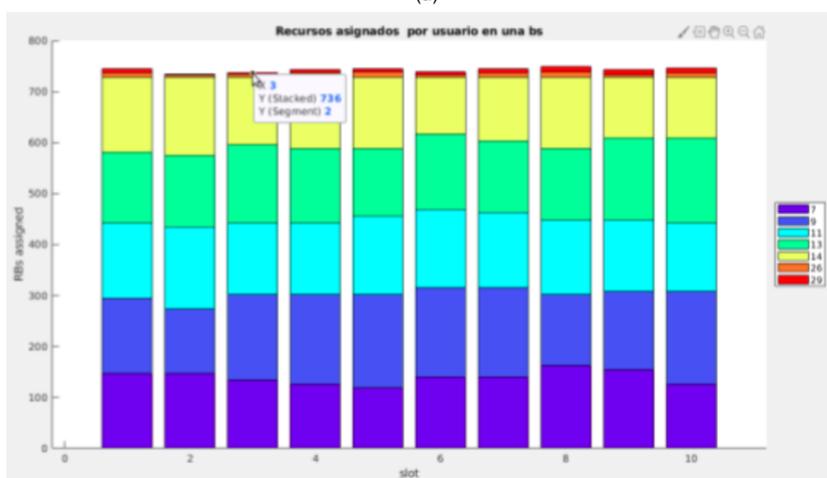
Caso HeU (Highest eMBB User)

Para obtener la simulación en esta modalidad se fijó el valor de γ en uno en el archivo "PropFairScheduler.m". Una vez distribuidos los usuarios en el escenario (como se ilustra en la Figura 7.14) se decidió trabajar con la BS 1 que atendía a

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing



(a)



(b)

Figura 7.10: Asignación de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en BR. (a) Usuario 26. (b) Usuario 29.

obj.rbGrid.DL.userAllocation								obj.rbGrid.DL.userAllocation							
	1	2	3	4	5	6	7		1	2	3	4	5	6	7
10	9	9	9	9	9	9	9	40	14	14	14	14	14	14	14
11	9	9	9	9	9	9	9	41	7	7	7	7	7	7	7
12	9	9	9	9	9	9	9	42	7	7	7	7	7	7	7
13	9	9	9	9	9	9	9	43	7	7	7	7	7	7	7
14	9	9	9	9	9	9	9	44	7	7	7	7	7	7	7
15	29	-1	-1	-1	-1	-1	-1	45	7	7	7	7	7	7	7
16	29	-1	-1	-1	-1	-1	-1	46	-1	26	-1	-1	-1	-1	-1
17	11	11	11	11	11	11	11	47	-1	26	-1	-1	-1	-1	-1
18	13	13	13	13	13	13	13	48	14	14	14	14	14	14	14
19	11	11	11	11	11	11	11	49	7	7	7	7	7	7	7
20	13	13	13	13	13	13	13	50	7	7	7	7	7	7	7
21	13	13	13	13	13	13	13	51	7	7	7	7	7	7	7
22	13	13	13	13	13	13	13	52	7	7	7	7	7	7	7
23	13	13	13	13	13	13	13	53	7	7	7	7	7	7	7

Asignación de recursos al usuario 29 (URLLC) en el primer minislot

Asignación de recursos al usuario 26 (URLLC) en el segundo minislot

Figura 7.11: Asignación de recursos al usuario 29 (URLLC) correspondientes al 1 minislot y al usuario 26 (URLLC) en el segundo minislot.

7.2. Proportional Fair con Puncturing

obj.rbGrid.DL.userAllocation							
	1	2	3	4	5	6	7
40	14	14	14	14	14	14	14
41	7	7	7	7	7	7	7
42	7	7	7	7	7	7	7
43	7	7	7	7	7	7	7
44	7	7	7	7	7	7	7
45	7	7	7	7	7	7	7
46	-1	26	-1	-1	26	26	-1
47	-1	26	-1	-1	26	26	-1
48	14	14	14	14	14	14	14
49	7	7	7	7	7	7	7
50	7	7	7	7	7	7	7
51	7	7	7	7	7	7	7
52	7	7	7	7	7	7	7
53	7	7	7	7	7	7	7

Figura 7.12: Asignación de recursos al usuario 26 (URLLC) correspondientes al quinto y sexto minislot.

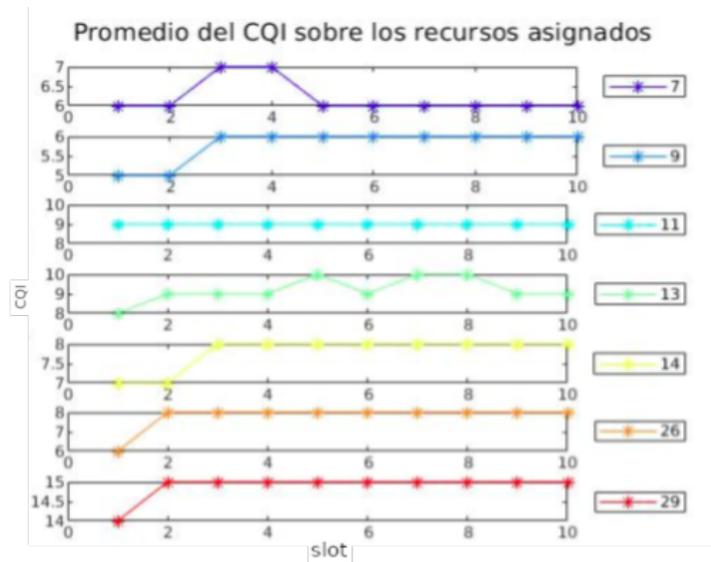


Figura 7.13: CQI Promedio de los RB asignados a los usuarios en la validación del Proportional Fair con Puncturing funcionando en BR.

tres usuarios eMBB (2, 12 y 18) y a dos usuarios URLLC (23 y 27). Se analizó el tmodel de los usuarios de baja latencia y se concluyó que el slot 4 resultaba interesante para ejemplificar el funcionamiento del algoritmo. En dicho slot el usuario 23 requirió 2 recursos en el minislot 3 y dos en el minislot 2 (total de 4 en el slot) mientras que el usuario 27 requirió 2 recursos en los primero tres minislots (6 RB en total). En la Figura 7.15 se observa que los requerimientos de los usuarios URLLC en el slot 4 fueron cubiertos.

Utilizando los breakpoints descritos anteriormente fue posible validar la forma en que se rellena la grilla de recursos. Se comienza a partir del momento en que se asignaron los recursos a los usuarios eMBB (5, 12 y 18). En el primer minislot el usuario 27 requirió dos RB para los cuales se pisaron los recursos 2 y 3 del usuario 18. Análogamente en el segundo minislot el usuario 27 requirió dos RB para los cuales se utilizaron los recursos 4 y 5 (también correspondientes al usuario 18). Esto se ilustra en la Figura 7.16 donde se aprecia que utilizando esta política no

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

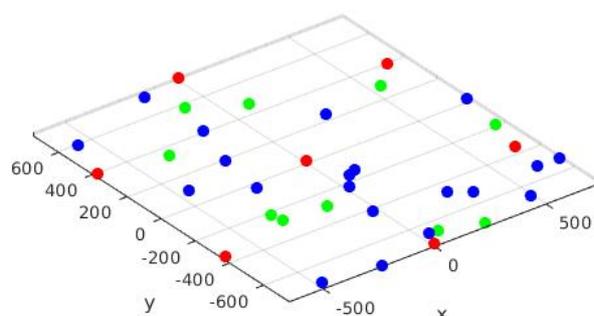


Figura 7.14: Escenario para la validación del Proportional Fair con Puncturing funcionando en HeU. Con rojo se representan las BS, con verde los usuarios URLLC y con azul los usuarios eMBB.

se reciclan los recursos que fueron pisados (como sucede en el caso BR).

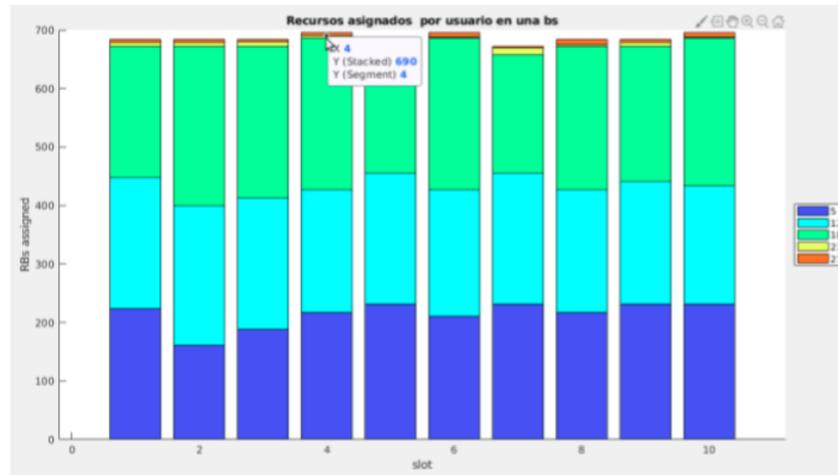
Prosiguiendo con el análisis, en el tercer minislot ambos usuarios requirieron recursos y en ambos casos se pisaron RBs del usuario 18. Al usuario 23 se le asignaron los recursos 45 y 46 y al usuario 27 se le asignaron los recursos 56 y 57. Por último, en el minislot 5 solamente el usuario 23 requirió recursos, para cubrir su demanda se le asignaron los RB 47 y 48 (también pertenecientes al usuario 18). El resumen de las asignaciones explicadas se visualiza en la Figura 7.17.

Para concluir la validación se puede observar en la Figura 7.18 que el usuario 18 es el que mejor CQI promedio tiene ($CQI = 8$), en comparación con los demás usuarios eMBB asignados en la BS (el usuario 5 tiene CQI promedio de valor 4 y el usuario 12 por su parte tiene CQI promedio 6). Esto es coherente en el resultado obtenido en la simulación que los usuarios URLLC pisaron los recursos del usuario 18 por ser el que tenía mejor CQI. Logrando obtener un promedio de CQI en sus recursos de 14 para el usuario 23 y de 7 en el usuario 27.

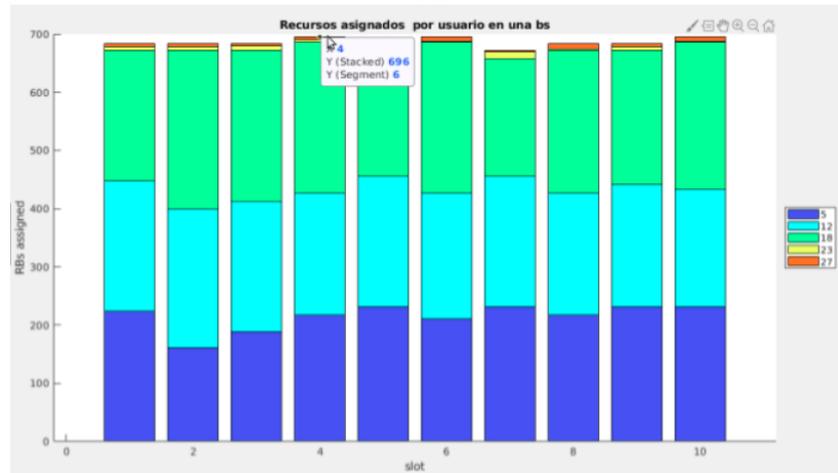
Caso LeU (Lowest eMBB user)

Se realizó una simulación en el mismo escenario planteado anteriormente (los usuarios y BS se distribuyeron como se muestra en la Figura 7.19) y se decidió trabajar con la BS 7 que atiende a los usuarios: 1, 4, 9 y 19 (eMBB) y a los usuarios 21 y 25 (URLLC). Se estudió el segundo slot de la simulación, en el cual los usuarios URLLC presentaban los siguientes requerimientos: el usuario 21 requirió dos RB en los minislots 1, 3, 5, 6 y 7 y el usuario 25 requirió dos RB en los minislots 1,4 y 6. En el primer minislot el usuario 21 pisó los recursos 48 y 49 del usuario 19 y por su parte el usuario 25 pisó los recursos 11 y 13 del usuario 4 (se visualiza en la Figura 7.20).

7.2. Proportional Fair con Puncturing



(a)



(b)

Figura 7.15: Asignación de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en HeU. (a) Usuario 23. (b) Usuario 27.

obj.rbGrid.DL.userAllocation								obj.rbGrid.DL.userAllocation							
	1	2	3	4	5	6	7		1	2	3	4	5	6	7
1	18	18	18	18	18	18	18	1	18	18	18	18	18	18	18
2	27	-1	-1	-1	-1	-1	-1	2	27	-1	-1	-1	-1	-1	-1
3	27	-1	-1	-1	-1	-1	-1	3	27	-1	-1	-1	-1	-1	-1
4	18	18	18	18	18	18	18	4	-1	27	-1	-1	-1	-1	-1
5	18	18	18	18	18	18	18	5	-1	27	-1	-1	-1	-1	-1
6	5	5	5	5	5	5	5	6	5	5	5	5	5	5	5
7	5	5	5	5	5	5	5	7	5	5	5	5	5	5	5
8	5	5	5	5	5	5	5	8	5	5	5	5	5	5	5
9	5	5	5	5	5	5	5	9	5	5	5	5	5	5	5
10	5	5	5	5	5	5	5	10	5	5	5	5	5	5	5
11	5	5	5	5	5	5	5	11	5	5	5	5	5	5	5
12	5	5	5	5	5	5	5	12	5	5	5	5	5	5	5

(a)

(b)

Figura 7.16: Asignación de recursos para el usuario 27 en la validación del Proportional Fair con Puncturing funcionando en HeU (a) Minislot 1 (b) Minislot 2.

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

obj.rbGrid.DL.userAllocation								obj.rbGrid.DL.userAllocation							
	1	2	3	4	5	6	7		1	2	3	4	5	6	7
44	18	18	18	18	18	18	18	44	18	18	18	18	18	18	18
45	-1	-1	23	-1	-1	-1	-1	45	-1	-1	23	-1	-1	-1	-1
46	-1	-1	23	-1	-1	-1	-1	46	-1	-1	23	-1	-1	-1	-1
47	18	18	18	18	18	18	18	47	-1	-1	-1	-1	23	-1	-1
54	18	18	18	18	18	18	18	48	-1	-1	-1	-1	23	-1	-1
55	18	18	18	18	18	18	18	49	18	18	18	18	18	18	18
56	-1	-1	27	-1	-1	-1	-1	50	18	18	18	18	18	18	18
57	-1	-1	27	-1	-1	-1	-1	51	18	18	18	18	18	18	18
58	18	18	18	18	18	18	18	52	18	18	18	18	18	18	18

Figura 7.17: Asignación de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en HeU (a) Minislot 3, usuarios 23 y 27 (b) Minislot 5, usuario 23.

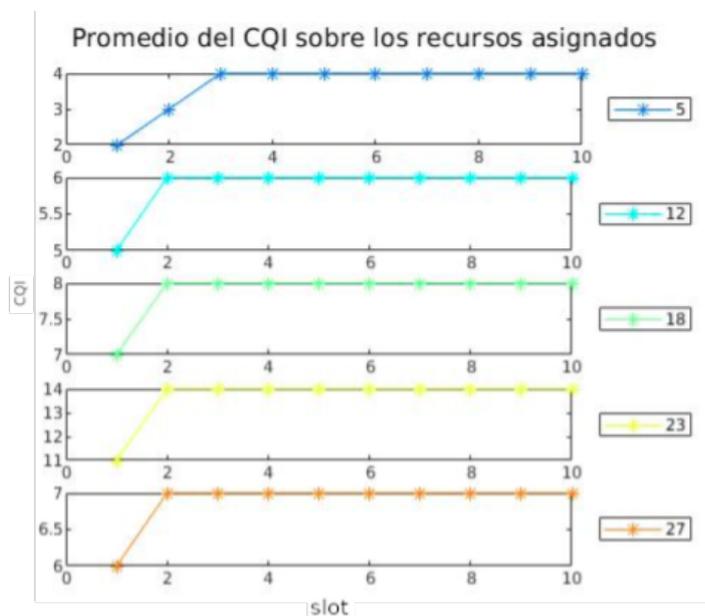


Figura 7.18: CQI Promedio de los RB asignados a los usuarios en la validación del Proportional Fair con Puncturing funcionando en HeU.

Al finalizar la simulación y obtener los resultados se comprobó que los recursos utilizados por los usuarios URLLC corresponden a los que tienen el CQI más bajo (como se visualiza en la Figura 7.21). El usuario 4 tiene un promedio de CQI de 4 y el usuario 19 tiene un promedio de CQI de 5 (ambos menores de 9 que es el promedio del CQI del usuario 9). El usuario 1 en el ejemplo no tuvo ningún recurso con CQI mayor a 0 (por eso no se toma en cuenta).

En la Figura 7.22 se visualiza la distribución final de los recursos asignados, comprobando que se repartieron todos los recursos requeridos por los usuarios URLLC. Se puede observar que en este caso también se da un reciclaje de los recursos pisados a usuarios eMBB (como sucedía en el caso BR). En este caso, una vez que se pisa un recurso eMBB cualquier usuario URLLC que requiera recursos en minislots posteriores puede usarlo. En la Figura 7.22 (a) se observa que el

7.2. Proportional Fair con Puncturing

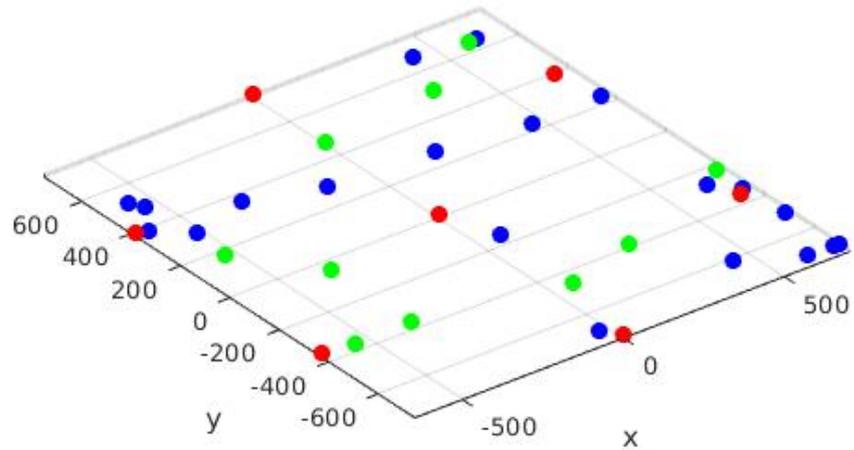


Figura 7.19: Escenario para la validación del Proportional Fair con Puncturing funcionando en LeU. Con rojo se representan las BS, con verde los usuarios ULLC y con azul los usuarios eMBB.

obj.rbGrid.DL.userAllocation								obj.rbGrid.DL.userAllocation							
	1	2	3	4	5	6	7		1	2	3	4	5	6	7
44								8	9	9	9	9	9	9	9
45	19	19	19	19	19	19	19	9	19	19	19	19	19	19	19
46	9	9	9	9	9	9	9	10	9	9	9	9	9	9	9
47	19	19	19	19	19	19	19	11	25	-1	-1	-1	-1	-1	-1
48	21	-1	-1	-1	-1	-1	-1	12	19	19	19	19	19	19	19
49	21	-1	-1	-1	-1	-1	-1	13	25	-1	-1	-1	-1	-1	-1
50	19	19	19	19	19	19	19	14	19	19	19	19	19	19	19
51	19	19	19	19	19	19	19	15	4	4	4	4	4	4	4
52	19	19	19	19	19	19	19	16	19	19	19	19	19	19	19
53	19	19	19	19	19	19	19	17	4	4	4	4	4	4	4
54	19	19	19	19	19	19	19	18	19	19	19	19	19	19	19
55	19	19	19	19	19	19	19	19	9	9	9	9	9	9	9

(a)

(b)

Figura 7.20: Asignación de recursos para los usuarios ULLC (en el primer minislot) en la validación del Proportional Fair con Puncturing funcionando en LeU. (a) Usuario 21. (b) Usuario 25.

Capítulo 7. Grilla con minislots y Proportional Fair con puncturing

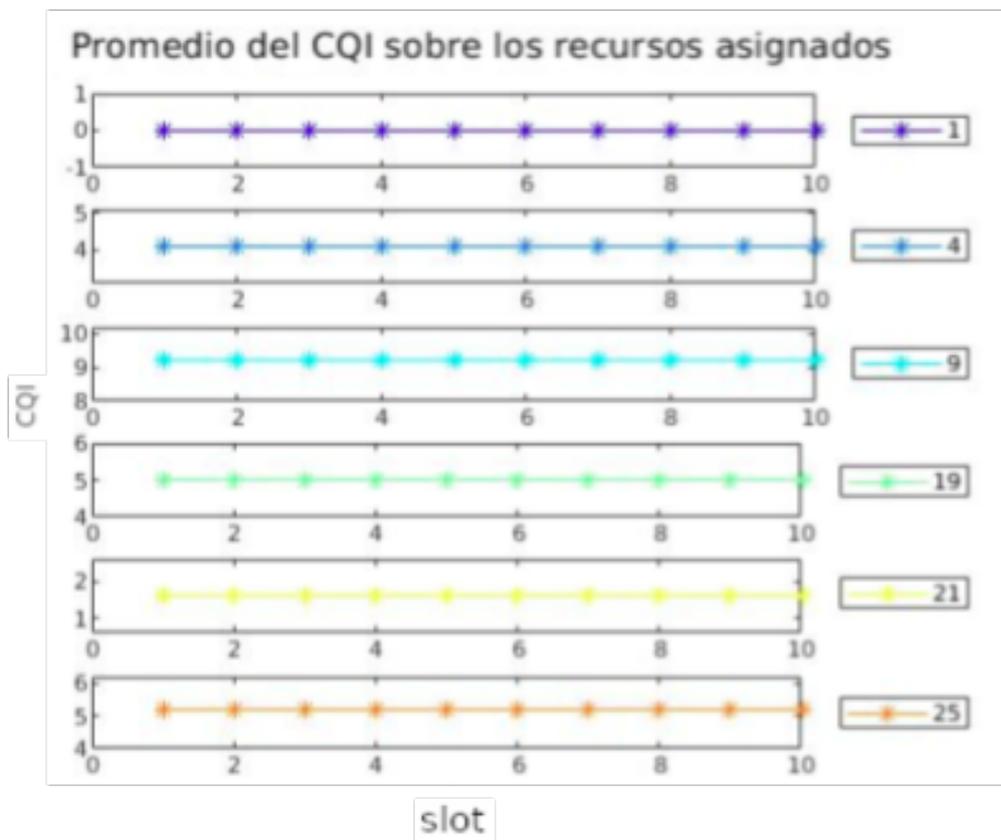


Figura 7.21: CQI Promedio de los RB asignados a los usuarios en la validación del Proportional Fair con Puncturing funcionando en LeU.

obj.rbGrid.DL.userAllocation								obj.rbGrid.DL.userAllocation							
	1	2	3	4	5	6	7		1	2	3	4	5	6	7
8	9	9	9	9	9	9	9	42	9	9	9	9	9	9	9
9	19	19	19	19	19	19	19	43	9	9	9	9	9	9	9
10	9	9	9	9	9	9	9	44	9	9	9	9	9	9	9
11	25	-1	21	25	21	21	21	45	19	19	19	19	19	19	19
12	19	19	19	19	19	19	19	46	9	9	9	9	9	9	9
13	25	-1	21	25	21	21	21	47	19	19	19	19	19	19	19
14	19	19	19	19	19	19	19	48	21	-1	-1	-1	-1	25	-1
15	4	4	4	4	4	4	4	49	21	-1	-1	-1	-1	25	-1
16	19	19	19	19	19	19	19	50	19	19	19	19	19	19	19
17	4	4	4	4	4	4	4	51	19	19	19	19	19	19	19
18	19	19	19	19	19	19	19	52	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	53	19	19	19	19	19	19	19

(a)

(b)

Figura 7.22: Asignación final de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en LeU.

recurso 11 se le concedió al usuario 25 en el primer minislot y posteriormente (en el tercero) se le dio al usuario 21. Esta alternancia no se da en el caso BR, donde cada usuario pisa los recursos que mejor satisfacen sus requerimientos.

A modo de concluir se destaca que esta política beneficia la transmisión eMBB y perjudica a los usuarios de baja latencia.

7.3. Resumen del Capítulo

En el presente capítulo se detalló la implementación y validación de la modificación de la grilla de recursos del simulador lo que permitió una estrategia para la implementación de minislots y el uso de puncturing. También se desarrolló la implementación y validación del algoritmo Proportional Fair con Puncturing. El mismo permite la convivencia de usuarios eMBB y URLLC abordando el problema de la asignación de recursos en dos partes (primero con los usuarios eMBB y posteriormente con los usuarios URLLC).

En el siguiente capítulo se detalla la implementación y validación del Framework desarrollado.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 8

Framework

Uno de los principales objetivos de este proyecto fue el desarrollo de un Framework para trabajar sobre algoritmos de scheduling en simuladores de redes NR. La finalidad de la herramienta es simplificar el proceso de diseño y simulación de algoritmos para la asignación de recursos, permitiendo construir una amplia variedad de algoritmos sin requerir de modificar el software del simulador y por lo tanto abstrayendo al usuario del funcionamiento interno del mismo. Además el Framework fue diseñado para no estar atado a un simulador particular, ya que lo que se modela es el proceso de asignación de recursos que realiza una radiobase, por esta razón es trasladable a cualquier simulador que contenga un scheduler (simuladores de sistema).

Basado en la comprensión del proceso de scheduling detallada en el Capítulo 3, el análisis de algoritmos y el proceso de diseño planteados en los Capítulos 6 y 7 y en los objetivos planteados para el Framework se procedió a realizar un modelado del proceso.

8.1. Modelado de scheduler

Como primer paso para el diseño del Framework fue necesario modelar un algoritmo de scheduling. Si lo observamos como un sistema con entradas y salidas, se puede modelar la entrada al scheduler como un conjunto de parámetros de la red y de cada usuario. La salida es la propia grilla con la asignación de recursos para cada usuario (portadora y potencia). Para modelar el sistema, se toman en cuenta los algoritmos trabajados en los Capítulos 6 y 7 (más concretamente Proportional Fair Generalizado y Best CQI) en los cuales la asignación se realiza de acuerdo al cálculo de una métrica por usuario para cada recurso, que dará un valor de prioridad para tomar dicho recurso. Dicha métrica es calculada en base a las entradas del sistema descriptas anteriormente. Luego para cada recurso se toma el usuario cuya métrica (prioridad) sea máxima y se le asigna el mismo.

Además el scheduler debe tener una forma de resolver en caso de que ocurra empate. Dicha decisión se puede tomar de distintas formas, entre ellas se puntualizan dos: decidir aleatoriamente y decidir en función de una cola de usuarios. La

Capítulo 8. Framework

cola se formará a medida que se enlacen usuarios a la radiobase, el usuario que entra va al final de la misma. Luego cuando a un usuario se le asigna un recurso se actualiza la cola pasándolo para el final de ella.

Teniendo en cuenta lo detallado en los párrafos anteriores se puede concluir que lo que diferencia un scheduler de otro se reduce a elegir de que forma se calcula la métrica y de que forma se decidirá en caso de haber un empate. Por lo tanto el Framework debe contar con ambas características para poder definir un scheduler (cálculo de la métrica y desempate). Para el cálculo de la métrica se optó por realizar el cálculo en varias etapas, en la Figura 8.1 se puede observar un diagrama del modelo utilizado. El bloque que se encuentra en la parte inferior (“Parámetros”) cumple la función de elegir de un listado predefinido de entradas al sistema, los parámetros de usuario que van a ser usados para el cálculo. A partir de los parámetros seleccionados se debe construir la métrica de cada usuario realizando una operación. Para realizar dicha operación se decidió que fuera en capas de operaciones básicas, las mismas se pueden colocar una sobre la otra para realizar operaciones más complejas. Cada capa posee una o varias salidas que serán las entradas de la siguiente. Como se puede observar en la Figura 8.1, la entrada de la primera capa son los parámetros seleccionados y la salida de la última es la métrica de cada usuario. Las capas tienen el objetivo de realizar operaciones entre sus entradas con el fin de combinarlas, por lo que deben ser operaciones que permitan realizarse entre varios términos. Además, los pesos que se dan a las entradas deben tener un neutro de tal forma que sea posible no tomar en cuenta una de ellas en la salida. Las características de las capas serán configuradas por medio de matrices y vectores.

Dado que puede ser de interés aplicar funciones sobre las entradas individuales (sin operar entre ellas), se agregan además las prefunciones (o prefunciones), estas son funciones que se aplican a cada entrada por separado con el objetivo de procesarlas antes de que ingresen a la operación de la capa, las mismas se muestran gráficamente en la Figura 8.1.

A continuación se detallan los tipos de capas elegidas para operar sobre los parámetros y se profundiza en el funcionamiento de las prefunciones.

8.1.1. Combinación lineal

Una de las operaciones básicas entre parámetros es la combinación lineal, la misma consiste en la suma de las entradas pesadas con coeficientes constantes. Dichos coeficientes son los que caracterizan la operación y por lo tanto la capa. Dado que el modelo elegido permite generar más de una salida por capa, para definirla se debe tener uno o más grupos de coeficientes para las distintas combinaciones lineales a generar. Los coeficientes ingresados pueden ser organizados en forma de matriz donde cada fila corresponde a un grupo de parámetros (fila i , grupo i) y cada elemento de la fila corresponde al coeficiente de la entrada (columna j , coeficiente de la entrada j). En la Ecuación 8.1 se muestra la forma de la matriz correspondiente a una capa con n entradas y m salidas.

8.1. Modelado de scheduler

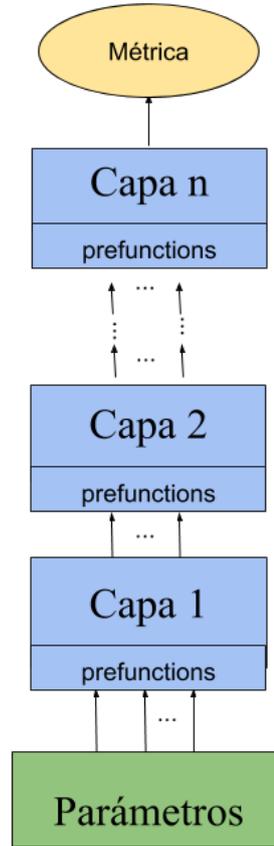


Figura 8.1: Diagrama del sistema utilizado para el cálculo de la métrica.

$$A = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,n} \\ C_{2,1} & C_{2,2} & \dots & C_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{m,1} & C_{m,2} & \dots & C_{m,n} \end{bmatrix} \quad (8.1)$$

Cada salida y_i se ve de la forma mostrada en la Ecuación 8.2.

$$y_i = \sum_{j=0}^n C_{i,j} \cdot x_j \quad (8.2)$$

Donde y_i es la salida i , x_j es la entrada j y $C_{i,j}$ los coeficientes que dan un peso a cada entrada. La operación se puede expresar de forma matricial como el producto de la matriz de pesos A con el vector de entradas X .

$$Y = A \cdot X = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,n} \\ C_{2,1} & C_{2,2} & \dots & C_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{m,1} & C_{m,2} & \dots & C_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad (8.3)$$

Capítulo 8. Framework

En caso de no querer tener en cuenta una entrada se utiliza el coeficiente 0, provocando que a la salida esa entrada no se encuentre en la sumatoria.

8.1.2. Producto con exponentes de peso

Otra operación que se consideró para los tipos de capas es la productoria junto con exponentes que representan el producto por si mismo de cada uno de los términos. Siguiendo el modelo de capas para lograr tener varias entradas y varias salidas, se expresan los exponentes dentro de una matriz en la cual cada fila corresponde a una salida y cada columna a una entrada. En la Ecuación (8.4) se muestra la forma de la matriz.

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{m,1} & b_{m,2} & \dots & b_{m,n} \end{bmatrix} \quad (8.4)$$

Cada salida queda por lo tanto expresada por la siguiente operación:

$$y_i = \prod_{j=0}^n x_j^{b_{i,j}} \quad (8.5)$$

Donde y_i es la salida i , x_j la entrada j y $b_{i,j}$ los coeficientes que dan un peso a cada entrada.

Una posible interpretación de dicha operación podría ser: elevar elemento a elemento la entrada con cada una de las filas por separado, obteniendo nuevos vectores. A dichos vectores se les hace la productoria de sus elementos y los resultados de dichas operaciones serán las salidas de la capa. Como se puede observar en la Ecuación (8.5), en este caso el coeficiente neutro será el 0 dado que al elevar la entrada nos dará 1 sin afectar al resto de la productoria.

Previo a operar en cada capa se da la posibilidad de aplicar “prefunciones” a las entradas. El funcionamiento de las mismas será explicado en la siguiente sección.

8.1.3. Prefunciones

Es de interés usar funciones que no pueden ser formadas a partir de las operaciones de la secciones anteriores (sumas y productos). Dada su naturaleza, no permiten operar entre términos (son funciones reales que se aplican a un valor) como lo son la exponencial, el logaritmo y las funciones trigonométricas. Ejemplos de algoritmos donde se usen estas funciones son ”Exponential/proportional fairness rule” donde se usa la exponencial y “Logarithm rule” que utiliza el logaritmo [56]. Las prefunciones atienden dicha necesidad dando la posibilidad de aplicar diferentes funciones a cada una de las entradas previo a ser ingresadas a una capa. En el modelo de capas propuesto las prefunciones se encuentran acopladas previo a la aplicación de la capa. Esto permite preprocesar las entradas aplicando una función

a cada una, además se incluye la identidad para el caso donde se busque que la entrada permanezca inalterada.

8.1.4. Desempate

Una vez calculada la métrica de todos los usuarios para un recurso, es necesario que el scheduler decida cuál es el usuario que tomará dicho recurso. Lo adecuado es elegir el usuario cuya métrica sea mayor. Existe la posibilidad de que más de un usuario posea el valor de métrica máximo por lo que es necesario realizar un desempate. Para ello el Framework plantea tomar la decisión basado en una política fija, dicha política puede ser:

- Aleatorio: El scheduler toma la decisión de desempate haciendo un sorteo entre los candidatos.
- Primero en la cola: El scheduler elige al usuario que se encuentre más adelante en la cola.
- Último en la cola: El scheduler elegirá al usuario que se encuentre más atrás en la cola.

Dicha cola es por radiobase y en ella se encuentran los usuarios enlazados a la misma. La cola se actualiza en 2 momentos, cuando se enlaza un nuevo usuario a la radiobase se lo agrega al final y cuando se le asigna un recurso a un usuario sin importar en que posición esté se lo mueve al final de la cola.

8.1.5. Modelado de ajustes

Para generar el algoritmo de scheduling de acuerdo al modelo detallado anteriormente es necesario ajustarlo. En principio se precisa seleccionar los parámetros que se utilizan para la métrica. Luego, para cada capa es necesario ingresar: el tipo de capa, la matriz correspondiente a dicha capa y además especificar las prefuncions a aplicarse previamente a cada entrada. Se puede organizar estos ajustes utilizando dos objetos: las “layers” o “capas” y el “framework scheduler” mostrados en la Figura 8.2.

El objeto “layer” contiene las prefuncions como un vector de strings por cada entrada indicando que función se le aplicará. Los otros parámetros son el tipo de capa y la matriz correspondiente a la capa (la cual debe respetar que la cantidad de columnas de la matriz debe ser igual a la cantidad de entradas). El objeto “framework scheduler” contiene los parámetros seleccionados para la métrica, utiliza un vector binario que indica con un “1” los parámetros a seleccionar dentro de un listado. También contiene un vector de capas que son del tipo “layer” y además una forma de desempatar los candidatos, por lo que se incluye un parámetro que puede ser una de las opciones disponibles: “first”, “last” o “random”.

A continuación se mostrará un ejemplo gráfico en el que se utilizarán las prefunciones y las capas detalladas anteriormente. Además se mostrará como podrían ingresarse los ajustes al Framework.

Capítulo 8. Framework

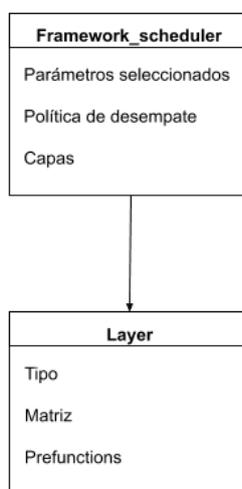


Figura 8.2: Diagrama de objetos implementados para el Framework.

8.1.6. Ejemplo de algoritmo

En la Figura 8.3 se puede ver un ejemplo gráfico de la generación de un algoritmo. En la parte inferior se muestra la selección de tres de los cuatro parámetros disponibles. Por encima se introduce una capa de combinación lineal llamada “S” (las prefunciones son la identidad), en la misma ingresan los tres parámetros seleccionados que pasan por la capa y tiene dos salidas, acordes con la cantidad de filas de la matriz. Dichas salidas ingresan en la siguiente capa la cual posee prefuncions que son aplicadas, para la primera entrada se aplica una exponencial y a la segunda no se la procesa (identidad). Luego las entradas procesadas ingresan en la capa de productoria “P”, donde son elevadas a los exponentes que se encuentran en la matriz y multiplicadas entre ellas. Cabe destacar que la salida es única dado que es la métrica.

Según el modelo de ajustes mostrado en la Subsección 8.1.5, para generar dicho scheduler con el Framework, los objetos de configuración son los mostrados en la Figura 8.4, suponiendo que se tomará como criterio de desempate el primero en la cola y como parámetros los primeros tres.

Como se puede observar a lo largo del diseño del Framework nunca se usó una funcionalidad específica de ningún lenguaje de programación o simulador específico, ya que las entradas y salidas del Framework corresponden al modelo de sistema de scheduling estudiado en la Sección 8.1. Para el cálculo de la métrica se utilizaron matrices y operaciones matemáticas independientes de cualquier software. Los ajustes fueron organizados en objetos o clases a modo de mantener un orden, pero dichos objetos no pertenecen a ningún lenguaje de programación en especial, incluso podrían organizarse los mismos ajustes de otra forma siempre y cuando se interpreten de la misma manera.

Con el objetivo de mostrar de que forma puede ser llevado a un simulador el Framework detallado y al mismo tiempo agregar valor al simulador, se llevará el

8.1. Modelado de scheduler

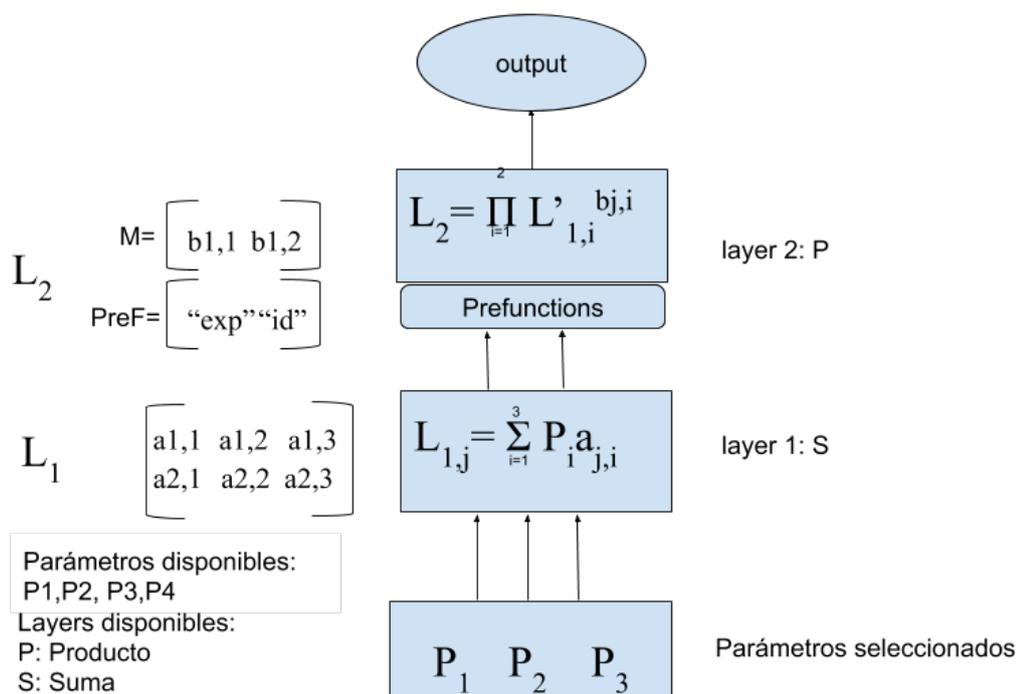


Figura 8.3: Diagrama de ejemplo del cálculo de la métrica.

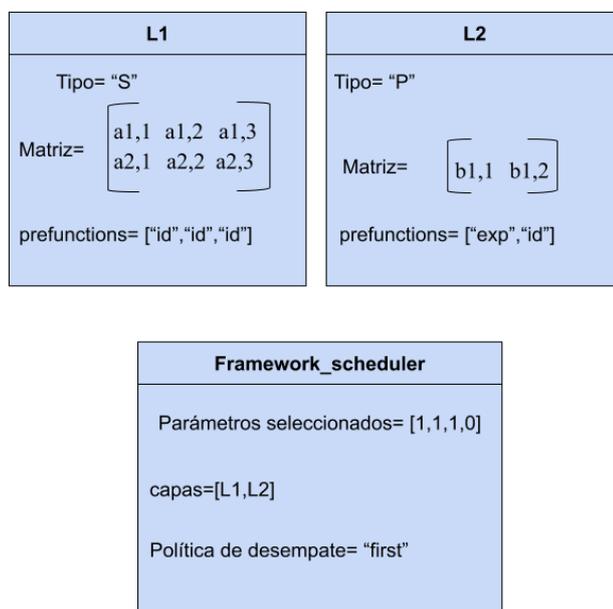


Figura 8.4: Objetos que contienen los ajustes para generar el scheduler de ejemplo.

Capítulo 8. Framework

modelo del Framework al simulador Vienna 5G SLS sobre lenguaje Matlab.

8.2. Implementación sobre el simulador Vienna 5G SLS

Basados en la estructura del simulador, que utiliza programación orientada a objetos, se implementaron los objetos donde se guardan los ajustes del algoritmo a crear. Los objetos contienen la estructura especificada en la Figura 8.2, para un Framework se debe crear un objeto “framework_scheduler”. A dicho objeto se le asignará una forma de desempate en “tiebreak_type” por medio de un string con alguna de las opciones: “first”, “last” o “random”. Se deben especificar los parámetros seleccionados por medio del vector binario tal como se explicó en la Subsección 8.1.5. Los parámetros de usuario disponibles en el Framework son los siguientes:

- CQI
- Bits transmitidos hasta el momento (ventana de tiempo de 20 slots por defecto)
- Cola de datos del usuario

El tercer parámetro que contiene las capas es el parámetro “layers”, a este se le debe asignar un vector ordenado de objetos “layer” correspondientes a las capas. Los objetos layer poseen los campos especificados en el modelo de ajustes de la Subsección 8.1.5. “type.layer” contiene el tipo de capa, las opciones de tipo son: “S” combinación lineal y “P” productoria. “matrix” contiene la matriz de la capa y prefuncions será un vector que contiene strings que especifican las prefunciones, las disponibles al momento son: “exp”, “log”, “cos”, “sin” y “id”. El constructor de la clase “framework_scheduler” contiene diferentes validaciones que detectan incompatibilidades. Por ejemplo verifica si las dimensiones de las matrices entre distintas capas concuerdan, la cantidad de filas de una capa (igual a la cantidad de salidas) debe ser igual a la cantidad de columnas de la matriz de la siguiente capa (cantidad de entradas). Otras validaciones verifican si los parámetros requeridos no están vacíos, y si se encuentran dentro de las opciones disponibles. Al detectar un problema imprimirá un error en consola mostrando los detalles del mismo y como solucionarlo. El objeto que contiene los ajustes debe ser guardado con nombre “fs” en un archivo .mat dentro de la carpeta “framework_scheduler_files” esto se puede hacer de forma fácil llamando al método “save_to_file” de la clase “framework_scheduler”. Un scheduler creado toma dichos ajustes y realiza la asignación de recursos (“FrameworkScheduler.m”). En el escenario se debe seleccionar “framework” como tipo de scheduler (explicación de escenarios en Sección 4.1.1) y especificar nombre del archivo .mat mediante el parámetro “filename”.

En el scheduler implementado se siguió una lógica similar a la utilizada en el diseño de los anteriores schedulers. En la Figura 8.5 se puede observar un diagrama de flujo del sistema implementado, el programa itera sobre todos los recursos calculando la métrica y buscando el máximo de las mismas, en caso de haber más de un candidato con la métrica máxima se hace el desempate. Luego se actualiza la cola pasando el usuario elegido al final de ella. Además se modifican los parámetros tales que la asignación haya afectado su valor (por ejemplo cantidad

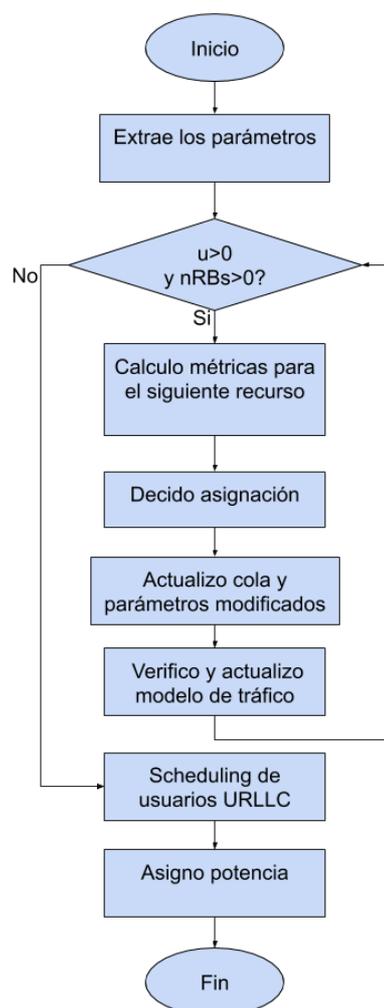


Figura 8.5: Diagrama de flujo para el Framework implementado. “u” representa la cantidad de usuarios cuya cola de datos no esta vacía y “nRBs” la cantidad de recursos restantes.

de bits a enviar). Posteriormente se realizan tareas relativas al control de tráfico, incluyendo verificar si algún usuario tiene satisfecha su cola de bits y actualizar el valor de la cola de datos del usuario. Al terminar el scheduling de los usuarios eMBB se prosigue con el scheduling de los usuarios URLLC (éste se realiza de la misma forma que en el Proportional Fair explicado en el capítulo anterior).

La parte central de este Framework es el cálculo de la métrica. Este bloque recibe los parámetros de cada usuario y toma el objeto que contiene los ajustes del Framework ingresados por el usuario. A partir de dicho objeto se toma de los parámetros disponibles solo los seleccionados por el usuario. Con dichos parámetros se arma una matriz de tamaño cantidad de usuarios por cantidad de parámetros. Teniendo la entrada a la primera capa, se procede a aplicar las capas una a una. Cada capa toma la matriz de entrada y a cada columna se le aplican las prefun-

ciones seleccionadas. Después que la matriz está procesada por las prefuciones se procede a aplicar la operación seleccionada (en el tipo de capa) que combina los parámetros.

8.3. Validación del Framework

Para validar el Framework se decidió realizar simulaciones con parámetros tales que generen algoritmos cuyos comportamientos son conocidos, por lo que se decidió comparar contra el Proportional Fair Generalizado. Más concretamente se reproducen los distintos casos de α y β analizados previamente (combinaciones de $\alpha = 1$ y $\alpha = 0$ con $\beta = 1$ y $\beta = 0$).

Se prueba primero un algoritmo que reparta de forma justa los bits, para eso se compara con el caso en el que $\alpha = 0$ y $\beta = 1$. En el Framework una forma de implementarlo es utilizando solo el parámetro bits transmitidos con una capa de producto con exponente -1. Se corrió una simulación, para realizar la comparación se utilizaron los mismos parámetros que en dicho capítulo y se tomo como referencia la radiobase 7, el valor de justicia medido mediante el índice de Jain nos da que la asignación es un 99,99 % justa, esto es acorde a lo esperado y a lo observado en el caso del Proportional Fair mostrado en el Capítulo 6. En las figuras 8.6 y 8.7 se puede observar que el comportamiento es bastante similar, intentando que el acumulado de bits sea igual para todos los usuarios. Por lo que se comprueba que el comportamiento del Framework y del parámetro es el esperado.

Se comparará ahora el Proportional Fair con $\alpha = 1$ y $\beta = 0$, contra un Best CQI implementado en el Framework. Para implementar dicho scheduler se utiliza únicamente el parámetro CQI y una capa identidad, en este caso se elige una capa de combinación lineal con coeficiente 1, esta capa no sumará ningún parámetro, únicamente armará la métrica con el CQI de cada usuario. Al simular con dichos ajustes en el Framework y tomar de ejemplo la radiobase 2, se observa que el índice de Jain de dicha asignación da un porcentaje de justicia del 18,03 %, esto es acorde con la baja justicia esperada y con lo obtenido previamente en el Capítulo 6. Observando la cantidad de recursos asignados a cada usuario (Figura 8.8), se puede identificar que a los usuarios con mayor promedio de CQI (Figura 8.9) son los que podrán recibir datos, dado los demás no tienen CQI que sea el máximo en ningún recurso. Este comportamiento caracteriza un Best CQI por lo que se concluye que el resultado del Framework al tomar el CQI es adecuado.

Se validará ahora la combinación de varios parámetros implementando un Proportional Fair utilizando el Framework, simulando y comparando contra el implementado sobre el simulador directamente (Capítulo 6). Para ello se seleccionan los parámetros CQI y bits transmitidos y se los divide con una capa de producto cuya matriz de exponentes es $[1, -1]$. Al hacer esto se asume que el CQI es un buen representante de la capacidad del canal, lo cual es lógico dado que a partir de él se decide el MCS a utilizar. Luego de simular tomando como ejemplo la radiobase 2, se obtuvo un porcentaje de justicia de 75,73 % para la asignación, un valor sensato dado que no se busca un algoritmo puramente justo, pero si que pesara la cantidad de bits transmitida por usuario. En la gráfica de bits a transmitir y el CQI pro-

Capítulo 8. Framework

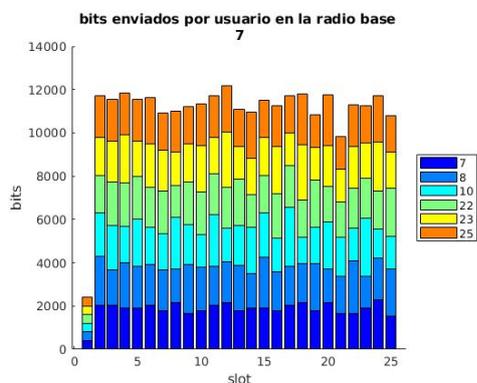


Figura 8.6: Bits enviados por usuario algoritmo justo implementado con el Framework.

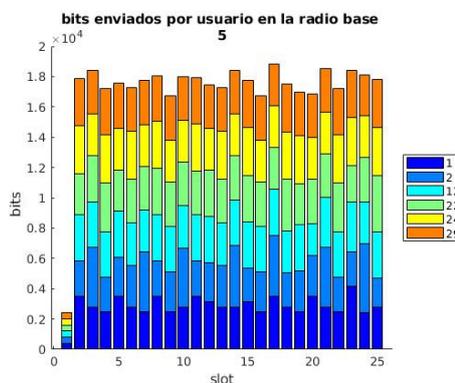


Figura 8.7: Bits enviados por usuario algoritmo Proportional Fair Generalizado con $\alpha = 0$ y $\beta = 1$.

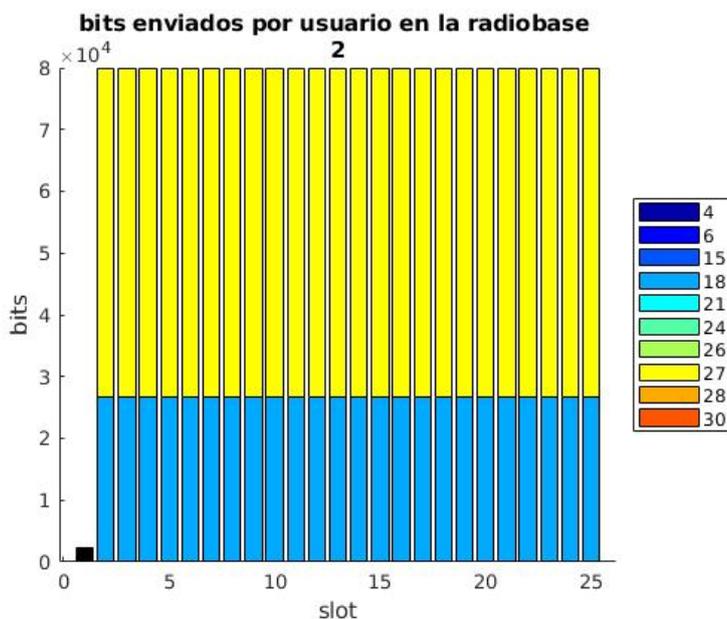


Figura 8.8: Recursos asignados por usuario utilizando el algoritmo best CQI implementado con el Framework.

medio de cada usuario se puede ver que los usuarios que logran transmitir mayor cantidad de bits son aquellos que tienen mayor CQI, pero sin dejar de transmitir aquellos usuarios cuyo CQI no es tan bueno. Este es el mismo comportamiento que el Proportional Fair Generalizado con $\alpha = 1$ y $\beta = 1$, además es acorde con lo esperado por lo que se concluye que la generación del algoritmo por parte del scheduler fue correcta.

8.3. Validación del Framework

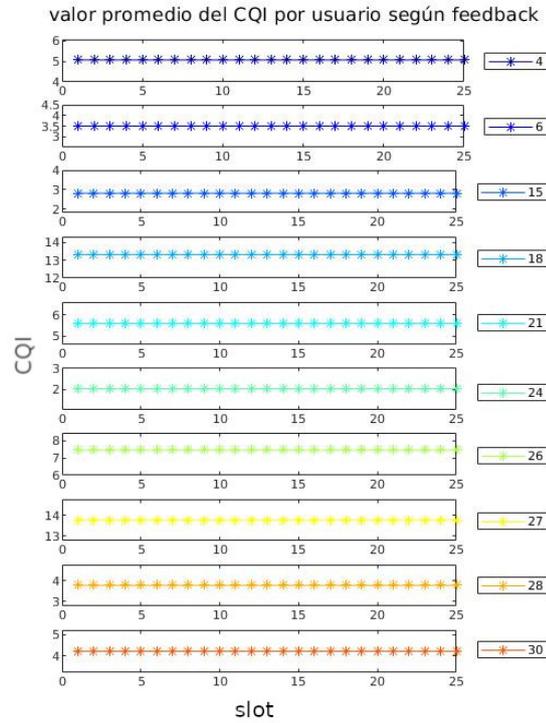


Figura 8.9: Promedio del CQI por cada usuario utilizando el algoritmo best CQI implementado con el Framework.

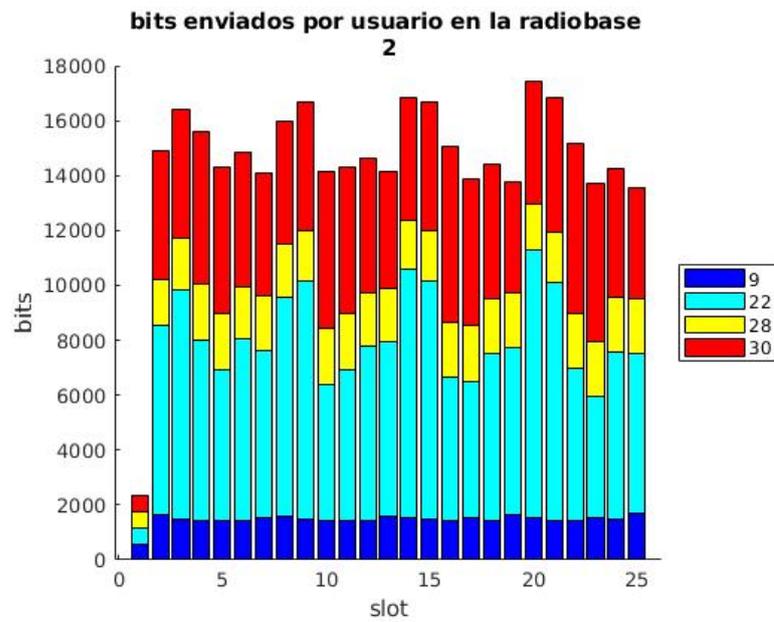


Figura 8.10: Recursos asignados por usuario utilizando el algoritmo Proportional Fair implementado con el Framework.

Capítulo 8. Framework

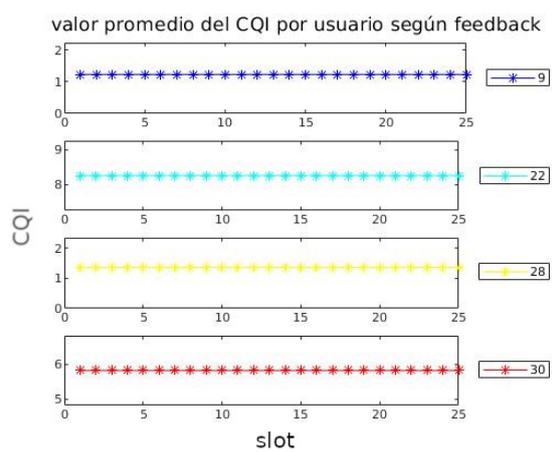


Figura 8.11: Promedio del CQI por cada usuario utilizando el algoritmo Proportional Fair implementado con el Framework.

8.4. Ejemplo de uso del Framework

El Framework permite crear una infinidad de schedulers, en el presente capítulo se validó el Framework diseñando un algoritmo justo, un Best CQI y Proportional Fair. A continuación se muestra la creación de un algoritmo que se ajusta a las necesidades de tráfico de dos tipos de usuarios diferenciados en la cantidad de tráfico que requieren recibir.

Se propone un algoritmo basado en el Proportional Fair donde se considere la cola de datos (Proportional Fair Queue). Por lo que se propone el índice

$$m_{i,k} = \frac{CQI_{i,k}}{Q_{i,k} \cdot T_{i,k}^2} \quad (8.6)$$

donde $m_{i,k}$, $Q_{i,k}$ y $T_{i,k}$ corresponden a la métrica, la cola de datos en kb y los kb transmitidos en los últimos 20 slots (incluyendo los que se asignaron al momento) para el usuario “i” en el recurso “k” respectivamente.

El objetivo de la métrica (8.6) es darle prioridad a los usuarios cuyo tráfico restante sea pequeño, de esta forma los usuarios que no tengan altos requerimientos de tráfico se verán beneficiados. Los usuarios cuyo comportamiento de tráfico se pueda modelar como pequeñas ráfagas, serán atendidos con mayor prioridad.

Para implementarlo en el Framework se utilizaron dos capas: la primer capa del tipo combinación lineal para pasar la cola y los bits transmitidos a kb (utilizando los coeficientes 1/1000) y la segunda capa de tipo producto para combinar los parámetros como división. En la implementación realizada sobre el simulador Vienna es posible generar dicho algoritmo, para ello se deben utilizar las siguientes líneas de código (para conocer en más detalle el uso del Framework ir al Apéndice E):

```
l1=layer( "S", [1,0,0;0,1/1000,0;0,0,1/1000], [ "id", "id", "id" ])
l2=layer( "P", [1,-2,-1], [ "id" , "id" , "id" ] );
fs = framework_scheduler([l1,l2], "first" , [1,1,1]);
fs.save_to_file("pf-queue");
```

Los parámetros utilizados en la métrica son el CQI, la cantidad de bits transmitidos y la cantidad de bits pendientes de transmitir (cola). En la primera línea se define la capa (layer) que contiene la cola y la cantidad de bits transmitidos multiplicados por 1/1000 para convertirlos a kilobits, y el CQI tal cual es recibido. El tipo de layer es “S” (combinación lineal) y la matriz es tal que las tres salidas son las tres entradas pero la cola y la cantidad de bits tendrán coeficiente 1/1000 y las prefuncions serán identidad. En la segunda línea se define la capa de producto donde se combinan los parámetros (cada uno con un exponente). El CQI tiene

Capítulo 8. Framework

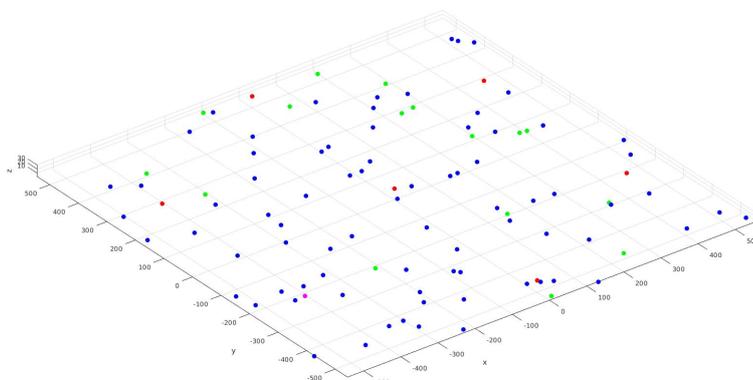


Figura 8.12: Posición de elementos de la red durante simulación de ejemplo para el Proportional Fair Queue.

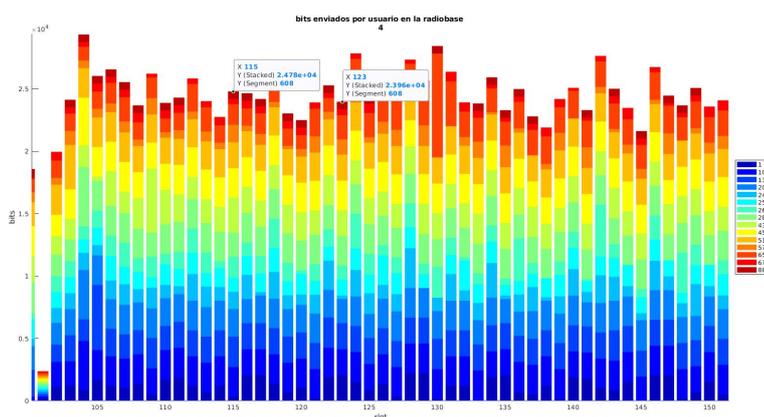


Figura 8.13: Bits transmitidos por usuarios en primeros 50 slots del chunk 2 en la radiobase 2.

exponente 1, la entrada kilobits transmitidos tiene exponente -2 (negativo ya que divide) y la cola de datos tiene exponente -1 . En la penúltima línea se crea el objeto framework scheduler que contiene un array con las dos layers, criterio de desempate primero en la cola (“first”) y se seleccionan los tres parámetros disponibles (“[1,1,1]”). Por último se lo guarda en un archivo con nombre “pf-queue”.

Se ejecutó la simulación con el escenario resumido en la Tabla 8.1, los usuarios fueron distribuidos de la forma mostrada en Figura 8.12, donde los usuarios azules son de alta demanda de tráfico y los verdes de baja. La radiobase 4 (color magenta $X = -400; Y = 200$) se tomó como ejemplo para observar la asignación de bits y recursos. En dicha radiobase se encuentra un único usuario (88) con ráfagas de 600 bits. Como se observa en la Figura 8.13 se cubren los requisitos de bits a pesar de su bajo CQI (Figura 8.14) en comparación con el resto de los usuarios. Por lo tanto se concluye que el algoritmo funciona como se esperaba en este tipo de escenarios.

8.4. Ejemplo de uso del Framework

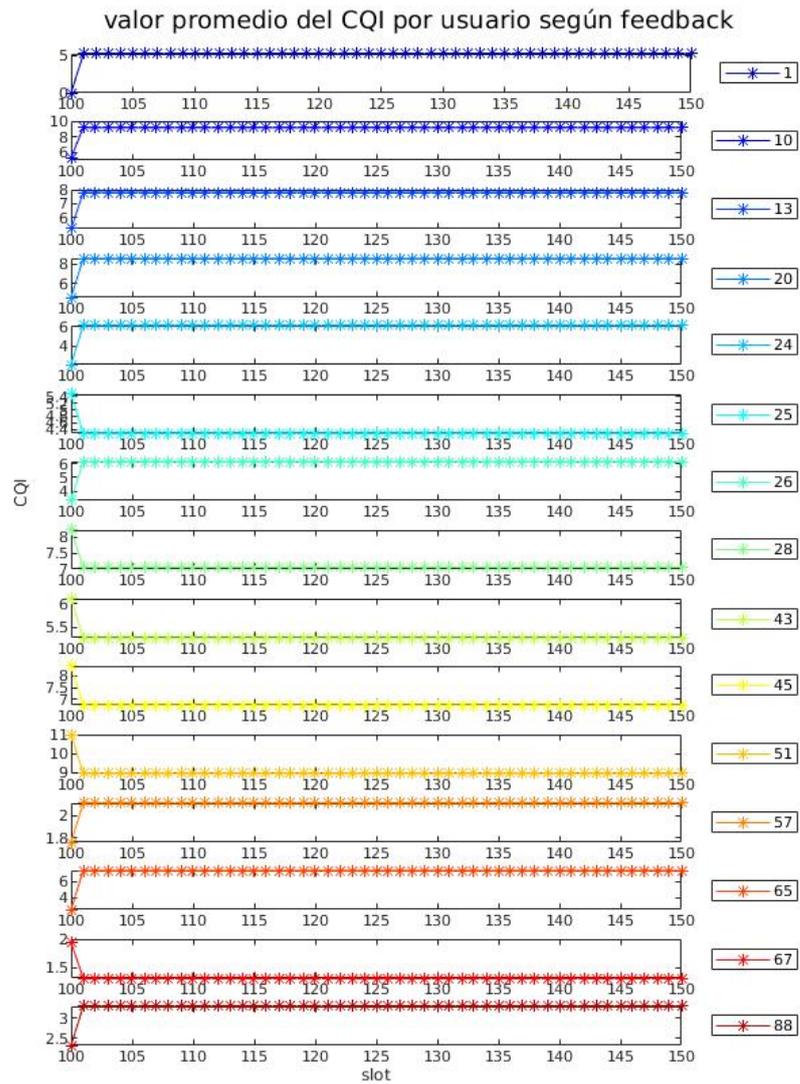


Figura 8.14: CQI por usuarios en primeros 50 slots del chunk 2 en la radiobase 2.

Capítulo 8. Framework

Tabla 8.1: Tabla resumen de escenario de simulación para probar el Proportional Fair Queue

Espacio			
Dimensiones	x=1100	y=1100	z=100
Obstáculos	Espacio abierto		
Tiempo			
Slots	100		
Chunks	3		
feedbackDelay	1		
Transmisión			
Ancho de banda útil para transmisión	98 %		
Overhead	14 %		
Ancho de banda de portadora	20MHz		
Duración del slot	1 ms		
Scheduler	Proportional Fair Queue		
Frecuencia central	2 GHz		
Radiobases			
Tipo	Macro		
Cantidad	7		
Antena	Omnidireccional		
Distribución	Hexgrid		
Usuarios			
Distribución espacial	Poisson		
Cantidad de usuarios (tráfico alto)	80		
Trafico de usuarios (tráfico alto)	Constante 2000 bits por slot		
Velocidad (tráfico alto)	20 km/h		
Cantidad de usuarios (tráfico en ráfagas)	15		
Velocidad (tráfico en ráfagas)	10 km/h		
Trafico de usuarios (tráfico en ráfagas)	Ráfagas (“burst”) de 600 bits		

8.5. Resumen del Capítulo

En el presente capítulo se detalla el modelo de la herramienta o Framework de generación de algoritmos de scheduling. Además se mostró una implementación sobre el simulador Vienna 5G SLS y se validó la misma generando algoritmos de comportamiento conocido y utilizando el Proportional Fair mostrado en el Capítulo 6 como referencia. Este capítulo culmina poniendo en práctica los conceptos del Framework, mostrando la practicidad y la capacidad que este tiene para generar nuevos algoritmos basados en métricas. Se mostró un posible algoritmo y las ventajas que puede llegar a tener en determinados escenarios. A continuación se desarrollan las conclusiones junto con los trabajos futuros que pueden desprenderse del presente proyecto.

Capítulo 9

Conclusiones y trabajo futuro

La tecnología 5G NR ya no es un horizonte lejano, se invierte cada vez más en el desarrollo de esta red móvil. Las características de la misma se enfocan en potenciar los diferentes servicios brindados actualmente como IoT o banda ancha e introduce la posibilidad de brindar nuevos servicios como los basados en URLLC. Estas novedades hacen que sea inminente la transición a esta nueva red. Debido a la revolución tecnológica que representa, se abren posibilidades de crecer en distintos campos que se encontraban limitados como es el caso del scheduling.

El presente proyecto se introdujo a la investigación dentro del área de scheduling sobre redes móviles y profundizó en la incidencia que tienen las características de 5G (como el soporte de servicios URLLC). Además brindó herramientas que contribuyen al desarrollo y diseño de nuevas soluciones.

9.1. Conocimientos adquiridos

Se logró investigar e incorporar el conocimiento relacionado a la tecnología 5G como la trama OFDM (posibilidad de utilizar múltiples espaciados de subportadoras), la estructura de la grilla de recursos que utiliza y la incorporación de servicios URLLC utilizando minislots. A partir de la comprensión de la red 5G se pudieron constatar las funcionalidades que carecía el simulador (por ejemplo 256 QAM). Estas carencias no influyeron en los resultados obtenidos ya que las implementaciones realizadas no dependen directamente de esto. No obstante, subsanar dichas carencias permitirá un desarrollo y expansión del simulador en el futuro junto con los posteriores releases de la 3GPP.

En cuanto al análisis y diseño de los distintos algoritmos de scheduling a lo largo del proyecto se logró una comprensión más profunda del funcionamiento y las características de cada uno, lo que permitió distinguir distintos factores de la red y del entorno que los afectan para saber cuándo es más adecuado utilizar uno u otro. Además el conocimiento de distintos algoritmos brindó herramientas para la generación de nuevos algoritmos o variantes de los ya conocidos, así como el Framework.

El estudio del scheduling como problema teórico aportó gran valor dado que

Capítulo 9. Conclusiones y trabajo futuro

permitió mejorar el diseño de algoritmos. Además de ser la base teórica para entender el funcionamiento de los schedulers, las herramientas que se expusieron fueron útiles para validar las implementaciones. Un ejemplo de esto es el índice de Jain, que se utilizó para medir la justicia en el caso del Proportional Fair. Utilizar las herramientas teóricas con fines prácticos concretos le aportó veracidad a los conceptos expuestos.

Otro resultado del proyecto fue la comprensión de los simuladores de redes móviles de sistema como el Vienna 5G SLS. De esta forma se adquirió manejo del software y facilidad para su modificación. Además el uso del mismo permitió encontrar las virtudes y defectos del simulador sirviendo de experiencia para investigaciones futuras.

9.2. Valor agregado introducido por el proyecto

Al comienzo del proyecto se supuso que el simulador Vienna 5G SLS 1.0 estaba en una etapa más avanzada con mayor cantidad de funcionalidades 5G. En cambio el simulador no contaba, entre otras novedades, con trama de duración variable, ni con minislots y además funcionaba en modalidad de tráfico full buffer. Los aportes realizados fueron de gran valor para cumplir con los objetivos propuestos. Si el simulador hubiera contado con estas funcionalidades se podrían haber explotado más profundamente las características de 5G en el modelado de los algoritmos de scheduler.

Entre los aportes realizados se encuentra el modelo de tráfico, el cual permitió modificar el comportamiento de los usuarios y dio lugar a la convivencia de usuarios con tráfico variable en el tiempo. De esta forma se adaptaron los escenarios de simulación a los distintos usos que se dan a la red. Implementar un modelo de tráfico aportó facilidades para la depuración en el desarrollo de los demás objetivos y permitió implementar los algoritmos de forma más completa. Le dio más realismo al simulador aunque se pueden seguir incorporando modelos de tráfico más complejos.

Otro de los objetivos del proyecto era la implementación de dos algoritmos sobre el simulador. Haber concretado dicho objetivo enriqueció el funcionamiento del simulador y le brindó al mismo más posibilidades para ofrecer a los usuarios que lo utilicen. El agregado de nuevos algoritmos dio más libertad al uso del simulador en distintos entornos, permitiendo comprender el impacto del scheduling en el comportamiento general de la red. La incorporación de los usuarios URLLC y el agregado de la técnica de puncturing permitió estudiar el impacto que tiene este servicio sobre el tráfico eMBB. La implementación de la grilla con minislots fue fundamental para lograr este objetivo. Esto permite un comportamiento de la red 5G más heterogéneo y acercado a la realidad.

Concretar el objetivo de la implementación del Framework en el simulador le agregó gran valor al mismo, dado que vuelve sistemática y simple la creación de un algoritmo de scheduling. El Framework permite que cualquier usuario del simulador pruebe sus propios algoritmos sin la necesidad de comprender en profundidad el código del simulador. Se mostró que a partir del Framework se pueden recrear

los demás algoritmos estudiados (Proportional Fair y Best CQI) y de forma muy sencilla se pueden obtener variantes de los mismos. Se logró desarrollar una herramienta fácil de utilizar que sintetiza la creación de algoritmos de asignación de recursos basados en una métrica. Como trabajo a futuro se plantea la incorporación de las novedades de 5G por encima del modelo planteado.

9.3. Trabajo Futuro

El presente proyecto da pie a la continuación del estudio e implementación de distintas ramas que se detallan a continuación.

Una de las más interesantes, dada la extensión actual, es el uso de IA (Inteligencia Artificial). Los algoritmos tratados durante el transcurso del proyecto toman la información del pasado y a partir de suponer que la red no varía demasiado asignan los recursos, dicha forma de decidir es estática. Sería de gran valor predecir el comportamiento futuro a partir del reconocimiento de patrones, de esta manera se lograría implementar algoritmos que aprendan de su entorno y se adapten, lo que favorecería a la toma de decisiones al momento de asignar recursos. Para diseñar estos algoritmos se debe ahondar en la teoría de IA aplicada al problema de la asignación de recursos.

Dentro del Framework existen diversas mejoras que se pueden hacer sin realizar grandes modificaciones gracias a la modularidad del modelo. Por ejemplo podría aumentarse la cantidad de parámetros disponibles para seleccionar o agregarse otros tipos de capas basados en operaciones distintas. También se encuentra la posibilidad de agregar formas más complejas de recorrer la grilla para asignar recursos.

Uno de los temas centrales del proyecto sobre el cual se basó el Framework y los estudios sobre el simulador fue la asignación de RBs, para lo cual se analizaron varios algoritmos y sus posibles implementaciones. Sin embargo no se tuvo en cuenta la asignación de potencia a los usuarios, lo cual también influye en el desempeño de la red. El estudio de distintos métodos de asignación de potencia puede ser de interés para investigaciones futuras.

Otro aspecto en el que es posible profundizar es en el modelado del tráfico, los modelos incluidos no refieren a ningún servicio de capa de aplicación. Es posible estudiar el comportamiento de distintos servicios que usen una red móvil e implementar modelos de tráfico que se acerquen al comportamiento de los mismos, de esta forma se aproxima aún más a la realidad la cantidad de tráfico que circula por la red en la simulación.

Una vez implementadas las funcionalidades faltantes en el simulador tales como: uplink, beamforming y network slicing se podrán extender los distintos schedulers así como el Framework, lo que permitirá el estudio del funcionamiento de estas tecnologías y de esta forma explotar las virtudes de las mismas. Además la implementación de puncturing realizada no utiliza la tabla correspondiente (Apéndice D) para el cálculo de throughput de usuarios URLLC, por lo que podría incluirse el soporte de múltiples tablas.

Dado que el modelado del Framework no se encuentra atado al simulador, el

Capítulo 9. Conclusiones y trabajo futuro

mismo puede ser extendido a otros simuladores de sistema. Esto permite elegir otro simulador a utilizar en caso de que el Vienna 5G SLS no se adapte a las necesidades de futuros estudios.

Apéndice A

Duplexación y acceso al medio

A.1. FDD y TDD

En FDD (Frequency Division Duplex) el transmisor y el receptor operan a diferentes frecuencias portadoras. Está separado el DL (DownLink) donde transmite la BS (Base Station) del UL (UpLink) donde transmite el móvil. En TDD (Time Division Duplex) la separación entre DL y UL lo da el instante de tiempo en que se usa el espectro.

A.2. Técnicas de acceso al medio

A.2.1. FDMA (Frequency Division Multiple Access)

El espectro disponible se divide en porciones de espectro más pequeñas (canales) y a cada usuario se le asigna un canal determinado para la comunicación. Fue una de las primeras técnicas utilizadas en los sistemas analógicos, por ejemplo AMPS (Advanced Mobile Phone System). Una de las desventajas que posee es que hace un uso ineficiente del espectro ya que dedica una frecuencia por usuario.

A.2.2. TDMA (Time Division Multiple Access)

La banda de frecuencias disponible es compartida por todos los usuarios, cada uno de ellos la usa en un tiempo finito. Dicho tiempo se lo conoce como TS (Time Slot). Es más eficiente que FDMA porque en una misma frecuencia, multiplexa en el tiempo a varios usuarios, esto hace aumentar la capacidad de FDMA. En contrapartida posee requerimientos críticos de sincronismo.

A.2.3. CDMA (Code Division Multiple Access)

Todos los usuarios transmiten en la misma porción de espectro y se diferencian por distintos códigos, esta técnica es utilizada en los sistemas de tercera generación, por ejemplo UMTS (Universal Mobile Telecommunications System). Es más

Apéndice A. Duplexación y acceso al medio

eficiente que TDMA pero al ser sumamente sensible a interferencia tiene grandes limitaciones con el manejo de la potencia.

Apéndice B

Red de telecomunicaciones Celular

B.1. Estructura de una red de telecomunicaciones

Las redes de telecomunicaciones se dividen en tres partes: red de acceso, red de transporte/distribución/agregación y núcleo de red (core).

B.1.1. Núcleo de red

Es el encargado de la intercomunicación dentro de la RAM (Red de Acceso Móvil) y hacia el resto de la red del operador u otros operadores. Provee interconexión para servicios de voz y de datos, aquí se encontraba la “inteligencia” de la red en las primeras generaciones, paulatinamente se va descentralizando hacia el acceso. El núcleo de red ha evolucionado junto con las tecnologías móviles: Core GSM (Global System for Mobile communications) 2G, UMTS (3G) y EPC (Evolved Packet Core) en LTE.

B.1.2. Red de transporte, distribución y agregación

La transmisión es el proceso de transportar información entre dos puntos de la red, en las telecomunicaciones, los sistemas de transmisión por lo general interconectan centrales telefónicas. Este es el segmento de la red que se encuentra entre el núcleo y la red de acceso, en ocasiones no se distingue del núcleo. Actualmente las señales son digitales y se utilizan regeneradores y multiplexores para transmitir múltiples señales por un medio físico.

B.1.3. Red de acceso

Es el segmento de la red de telecomunicaciones que conecta a los usuarios finales con la red del proveedor (con el concentrador de la central). En el caso de la telefonía móvil la red de acceso es inalámbrica, lo que hace que sea una red más barata de instalar frente a una red de acceso cableada (cobre o fibra óptica). La red de acceso móvil se conoce también como “celular” pues se implementa el concepto de celda para nombrar el área de cobertura de un sector (grupo de antenas). Una

Apéndice B. Red de telecomunicaciones Celular

celda se presenta como un hexágono para simplificar la cobertura ideal. La BTS (Base Transceiver Station) o radiobase o BS (Base Station) o Nodo B cuenta con transmisores, combinadores, equipos de Transmisión y pueden tener el control de más de una celda.

B.2. Outdoor de una radiobase

Los elementos que componen el outdoor de una radiobase son los feeders, antenas, TMA (Tower Mounted Amplifier), jumpers y conectores. Para configurar el outdoor se debe tener en cuenta las siguientes características: ganancia de la antena, acimut (dirección de máxima ganancia de la antena), apertura horizontal de la antena, apertura vertical. El tilt (eléctrico y mecánico) que delimita el alcance de la celda, la polarización vertical o polarización cruzada y la relación frente espalda.

B.3. Clasificación de celdas

B.3.1. Según cantidad de sectores

Existen las antenas Omnidireccionales, que presentan una cobertura en toda la celda, cobertura de 360° y azimut 0° . Y celdas sectorizadas que presentan de 2 a 6 sectores. Siendo las más comunes las de 3 sectores, con antenas que cubren 120° por celda y tienen acimut $0^\circ/120^\circ/240^\circ/$.

B.3.2. Según su alcance, potencia de transmisión y área de cobertura

Existen “pico-celdas” que tienen un área de cobertura indoor, “micro-celdas” con áreas de cobertura de decenas o cientos de metros (hasta 1km) y “macro-celdas” con cobertura desde unos pocos kilómetros a 35km como en el caso de sistemas GSM.

B.4. Evolución de la telefonía celular

B.4.1. Telefonía Móvil de Primera Generación AMPS (1G)

Las redes de telefonía móvil de primera generación datan de las décadas de los 70' y los 80', eran sistemas basados en técnicas de transmisión analógica, con baja eficiencia espectral y sin seguridad. Los sistemas eran incompatibles (sin posibilidad de roaming) y contaban solamente con servicios de voz de baja calidad. La tecnología predominante fue AMPS (Advanced Mobile Phone System), implementado por primera vez en los laboratorios Bell en 1983. El sistema utiliza celdas que dividen el espacio geográfico y FDMA (Frequency Division Multiple Access) que asigna distinto número de frecuencias para cada comunicación establecida.

B.4. Evolución de la telefonía celular

Fue un servicio de voz analógica con varios inconvenientes, entre otros el handover (transferencia de una celda a otra).

B.4.2. Telefonía Móvil de Segunda Generación GSM (2G)

Los sistemas de segunda generación fueron digitales, permitieron mejorar la calidad del servicio y ofrecieron mayor capacidad para servicios de voz (eficiencia espectral), además aportaron seguridad a las comunicaciones, implementando autenticación de usuario y cifrado. Son sistemas de segunda generación IS-136 (TDMA, Time Division Multiple Access), IS-95 (CDMA, Code Division Multiple Access) y GSM (Global System for Mobile Communications) surgida a comienzos de los 90' en Finlandia, resultó ser la tecnología que se impuso en la telefonía móvil de segunda generación.

GSM introdujo mejoras en la eficiencia espectral respecto a su precursor AMPS, al ser estándar ETSI [57] (European Telecommunications Standards Institute) y poseer servicios de roaming se facilitó su globalización. También posibilitó la separación del equipo y el usuario con la implementación de SIM (Subscriber Identity Module), estas cualidades posicionaron a GSM como líder en el mercado y marcó un antes y un después en las comunicaciones móviles ya que esta tecnología permitió que el servicio se masificara a nivel global. Con la introducción de GPRS/EDGE (General Packet Radio Service - Enhanced Data Rates for GSM Evolution), considerado 2.5G, se comenzó a ofrecer servicios orientados a paquetes utilizando la red de acceso GSM ya existente, lo que facilita el despliegue. Los bitrates alcanzados eran de alrededor de 100 kbps, suficientes para utilizar servicios de correo electrónico y acceder a páginas web.

B.4.3. Telefonía Móvil de Tercera Generación UMTS (3G)

Los sistemas de tercera generación fueron orientados a servicios de datos, ofreciendo banda ancha móvil, video call, streaming, lograron mayor eficiencia espectral que 2G/2.5G y las técnicas de acceso al medio que implementó fueron WCDMA (Wide Code Division Multiple Access) y OFDMA (Orthogonal Frequency-Division Multiple Access).

El estándar UMTS (Universal Mobile Telecommunications System) se convirtió en el más representativo de una red móvil de tercera generación, logró mejorar los bitrates de los servicios de conmutación de paquetes, pasando de 384 kbps, hasta superar los 10 Mbps a la vez que mantiene los servicios de voz. A nivel de radio cambia completamente el acceso al medio, en GSM se utilizaba la tecnología FDMA/TDMA mientras que en UMTS se implementa WCDMA, estandarizado por 3GPP en 1999 [1].

En la actualidad permanecen vigentes las tecnologías UMTS, CDMA2000 (Code Division Multiple Access, interfaces aprobadas por el estándar ITU IMT-2000), WiMAX (Worldwide Interoperability for Microwave Access), tecnologías HSPA (High-Speed Packet Access) permitieron versiones de 3.5G y 3.75G como el caso HSPA+(MIMO, Multiple-Input Multiple-Output), que dio paso a LTE.

B.4.4. Telefonía Móvil de Cuarta Generación LTE (4G)

Debido a las limitaciones de diseño en el estándar UMTS, el grupo 3GPP junto con la asociación de fabricantes de dispositivos móviles, desarrolladores de infraestructura y operadores de redes móviles decidieron rediseñar la red de radio y el core de la red, a partir de allí nace “LTE (Long Term Evolution)” como la evolución de UMTS. Los primeros trabajos sobre LTE comenzaron a finales del 2004, con el propósito principal de proveer una nueva RAT (Radio-Access Technology) centrada solamente en la conmutación de paquetes. La primer versión de las especificaciones de LTE (versión N° 8 [14]) fue terminada en 2008 y los operadores comenzaron a comercializarlo a principios del 2009. También se introduce, en paralelo con la red de acceso, un nuevo núcleo de red llamado EPC (Evolved Packet Core). La décima versión [16] es la primera que incluye LTE-Advanced Pro, mientras que las versiones 14 [22] y 15 [23] también incluyen evoluciones de LTE como se puede ver en la figura B.1.

Uno de los principales avances fue la introducción de OFDM (Orthogonal Frequency Divison Multiplex) que consiste en la división del espectro en múltiples portadoras con frecuencias ortogonales. Se divide el ancho de banda de la portadora (20MHz) en 1200 subportadoras de 15kHz que transmiten de forma simultánea de esta manera se mitigan los problemas relacionados con el multicamino^I que presentaba WCDMA. El otro gran cambio fue la introducción de la tecnología IP, en UMTS se utilizaba la tecnología de conmutación de circuitos, con la introducción de la conmutación de paquetes se simplifica la RAN (Radio Access Network), la red de transporte y el núcleo de la red.

B.4.5. Descripción general de la primer versión

Un requerimiento importante que se le impuso al desarrollo de LTE fue la flexibilidad espectral, un ejemplo de esto es la capacidad para soportar TDD y FDD. El esquema de modulación OFDM permite explotar el dominio del tiempo y de la frecuencia además de contar con una baja complejidad para el receptor y aportar robustez, de esta forma aprovecha también otra cualidad básica de LTE que es la multiplexación espacial MIMO (Multiple Input - Multiple Output).

Como el diseño de LTE fue inicialmente orientado hacia redes macro con frecuencias de portadoras de hasta unos pocos Ghz, se decidió utilizar un único espaciado de subportadoras de 15 kHz y un prefijo cíclico de 4.7 μ s^{II}. En el dominio del tiempo LTE organiza las transmisiones en tramas de 10 ms, consistiendo cada una en diez subtramas de 1 ms correspondiendo a 14 símbolos OFDM (es la mínima unidad de asignación de recursos en LTE). El slot en LTE se conforma por 7 símbolos OFDM y el RB se conforma entonces por 2 slots o 14 símbolos OFDM [49] (lo mismo se ilustra en la Figura B.2).

^ILas señales de radio llegan a las antenas receptoras por diferentes caminos, cada uno de un largo particular, con la misma velocidad de propagación en todos los caminos lo que provoca que lleguen en diferentes tiempos [58]

^{II}Existe la opción de un prefijo cíclico extendido de 16.7 μ s, aunque rara vez se usa en la práctica.

B.4. Evolución de la telefonía celular

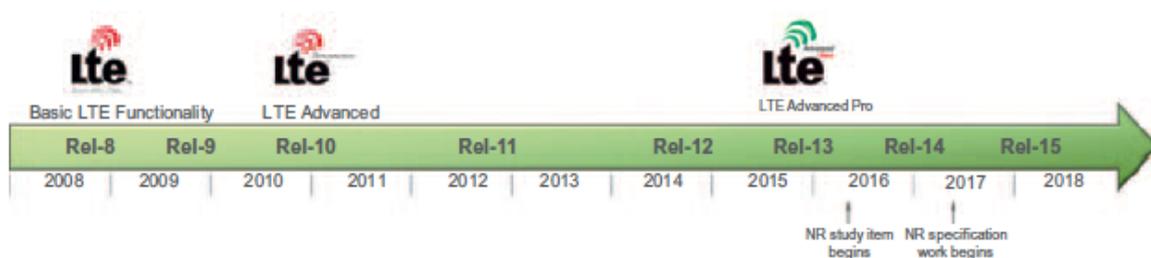


Figura B.1: Evolución de LTE [17].

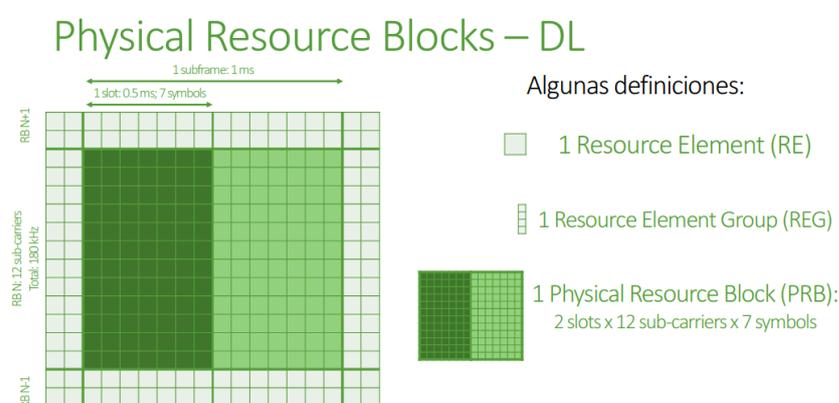


Figura B.2: Tamaño del slot y PRB en downlink LTE [49].

Dado los escenarios hacia los que está orientado LTE: celdas relativamente extensas con muchos usuarios en cada una, la BS transmite continuamente una o mas señales de referencia (sin importar si hay información para transmitir o no). Estas señales de referencia son usadas por varias funciones de LTE: estimar la respuesta al impulso del canal para una demodulación coherente, reportes del estado del canal para decisiones de scheduling, corrección de errores de frecuencia y medidas para dar información de movimiento de los usuarios. La señal de referencia depende de la cantidad de antenas transmitiendo simultáneamente en una celda.

Para el caso usual de 2x2 MIMO se utilizan 4 símbolos OFDM de los 14 existentes en una subtrama, con una distancia de 3 subportadoras en frecuencia, para transmitir las señales de referencia. Entonces en el dominio del tiempo hay alrededor de $200\mu\text{s}$ entre llegadas de una señal de referencia, lo cual limita las posibilidades de apagar el transmisor para reducir el consumo de energía. La transmisión de datos en LTE es programada principalmente en forma dinámica tanto en los enlaces de subida como de bajada y para aprovechar las condiciones, que suelen variar rápidamente, se pueden usar schedulers con métricas que dependen del canal. Para cada subtrama de 1 ms el scheduler controla qué dispositivos transmiten o reciben y en qué frecuencias, también se pueden seleccionar distintas tasas tanto ajustando la tasa del FEC (Forward Error Correction) encoder como variando el esquema de modulación entre QPSK, 16-QAM y 64-QAM [17]. Para

Apéndice B. Red de telecomunicaciones Celular

manejar los errores de transmisión LTE emplea HARQ (Hybrid Automatic Repeat Request [59]) con soft combining a través del cual el dispositivo indica, tras la recepción, el resultado de la decodificación a la BS, la cual puede retransmitir bloques recibidos erróneamente. Soft combining no descarta los bloques recibidos erróneamente, sino que los guarda y lo combina con el bloque retransmitido para bajar la probabilidad de que la retransmisión también tenga errores.

Las decisiones del scheduler son entregadas al dispositivo a través del PDCCH (Physical Downlink Control Channel), si hay más de un dispositivo programado en la misma subtrama, lo cual es común, entonces hay varios PDCCHs, una por cada dispositivo programado. Los primeros tres símbolos OFDM de la subtrama son usados para transmitir los canales de control del enlace de bajada, cada canal de control abarca todo el ancho de banda del espectro, maximizando la diversidad de frecuencias. Esto también implica que todos los dispositivos deben soportar la banda completa del espectro hasta el valor máximo de 20MHz. Las señales de control del enlace de subida por los dispositivos, por ejemplo los ACK (ACKnowledge), HARQ y la información del estado de canal para la programación del enlace de bajada, es transportada sobre el PUCCH (Physical Uplink Control Channel) el cual tiene una duración base de 1 ms.

Apéndice C

Entes reguladores, organizaciones de estándares y foros industriales

La investigación, desarrollo, implementación y despliegue de los sistemas de comunicaciones móviles es realizado por la industria inalámbrica en un esfuerzo internacional coordinado, mediante el cual se acuerdan especificaciones comunes de la industria que definen el sistema completo de comunicación móvil. El trabajo depende fuertemente de las regulaciones globales y regionales, en particular para el uso del espectro el cual es un componente esencial para todas las tecnologías de radio. Hay diferentes organizaciones involucradas en crear especificaciones técnicas y estándares como también en regular el área de las comunicaciones móviles, esto se puede dividir en tres grupos: organizaciones desarrolladoras de estándares (SDOs - Standards Developing Organization), cuerpos reguladores y administrativos y foros industriales [60].

Las SDOs desarrollan y discuten sobre estándares y especificaciones técnicas para sistemas de comunicaciones móviles, tienen el fin de hacer posible el despliegue de productos estandarizados para la industria y proveer interoperabilidad entre dichos productos. La mayoría de los componentes de los sistemas de comunicaciones móviles, incluyendo las radio bases y los dispositivos móviles (teléfonos celulares), son estandarizados hasta cierto punto.

Si bien existen SDOs nacionales, debido a la extensión global de los productos de comunicaciones, la mayoría de las SDOs son regionales y también cooperan a nivel global, por ejemplo, las especificaciones técnicas de GSM, WCDMA/HSPA, LTE y NR han sido todas creadas por la 3GPP (Third Generation Partnership Project) [1]. La 3GPP es una organización global formada por siete SDOs regionales y nacionales de Europa (ETSI [57]), Japón (ARIB -Association of Radio Industries and Businesses- y TTC -Telecommunications Technology Commitee-), Estados Unidos (ATIS -Alliance for Telecommunications Industry Solutions-), China (CCSA -China Communications Standards Association-), y la India (TSD-SI -Telecommunications Technology Development Society, India-) [60]. Las SDOs tienen un grado de transparencia variable, 3GPP, por ejemplo, es completamente transparente con todas las especificaciones técnicas, documentos de reuniones, reportes, y reflectores de correos electrónicos disponibles públicamente además de

Apéndice C. Entes reguladores, organizaciones de estándares y foros industriales

libre acceso, incluso para organizaciones que no forman parte de 3GPP.

Los entes reguladores y administrativos son organizaciones lideradas por cada gobierno que ponen requerimientos legales y de regulación para vender, desplegar, y operar sistemas de comunicaciones móviles y otros productos de telecomunicaciones. Una de las tareas más importantes es controlar el uso del espectro y establecer condiciones de licencia para los operadores móviles que usan partes del espectro de RF (Radio Frecuencia) para operaciones móviles. La regulación del espectro es manejada tanto a nivel nacional por administraciones nacionales como por cuerpos regionales: en Europa (CEPT -European Conference of Postal and Telecommunications Administration-/ECC -Electronic Communications Committee-), en América (CITEL -Inter-American Telecommunication Commission-), y Asia (APT) [60]. A nivel global, la regulación del espectro es manejada por la UIT (Unión Internacional de Telecomunicaciones o ITU en inglés).

En Uruguay la URSEC (Unidad Reguladora de Servicios de Comunicaciones) es la encargada de regular, controlar y promover las actividades de telecomunicaciones y postales con transparencia, eficiencia, reconociendo las máximas garantías procedimentales, la efectividad de los derechos en la materia y promoviendo el desarrollo en sus dimensiones económicas, social, cultural y ambiental, velando por la competencia dentro de los límites establecidos por el derecho Uruguayo, en beneficio de la Sociedad. Algunos de los cometidos que tiene la URSEC son asesorar al Poder Ejecutivo y a sus organismos competentes aportando insumos para la formulación, instrumentación y aplicación de la política de telecomunicaciones, velar por el cumplimiento de las normas sectoriales específicas y administrar, defender y controlar el espectro radioeléctrico nacional [61].

Los foros industriales son grupos liderados por la industria, promoviendo y dando lugar a tecnologías específicas u otros fines. Si bien estos grupos son liderados por los operadores, también hay fabricantes y empresas creando foros industriales. Por ejemplo GSMA (GSM Association), la cual está promoviendo tecnologías de comunicaciones móviles basadas en GSM, WCDMA, LTE y NR, NGMN (Next Generation Mobile Networks). GSMA es un grupo de operadores que define requerimientos sobre la evolución de los sistemas móviles, así como “5G Américas” el cual es un foro industrial regional que ha evolucionado de su predecesor “4G Americas” [60]. La figura C.1 muestra gráficamente la relación entre diferentes organizaciones involucradas en establecer regulaciones y limitaciones técnicas para los sistemas móviles, las flechas continuas indican documentación formal como estándares técnicos, recomendaciones y mandatos de reguladores que definen las tecnologías y la regulación, las flechas punteadas muestran participaciones indirectas.

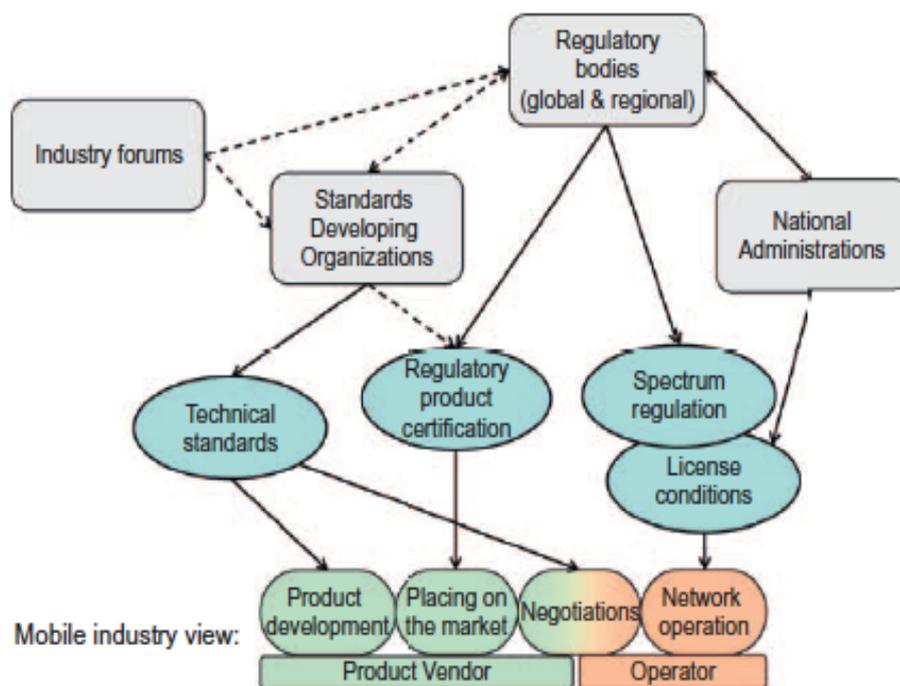


Figura C.1: Visión gráfica y simplificada de la relación entre las SDOs, cuerpos reguladores y administrativos y foros industriales [60].

Esta página ha sido intencionalmente dejada en blanco.

Apéndice D

Tablas CQI para LTE y 5G

Tabla D.1: Tabla para la elección de parámetros de acuerdo a CQI para $BLER < 0,00001$, extraída de [23], Tabla 5.2.2.1-4

CQI Index	modulation	code rate X 1024	efficiency
0	out of range		
1	QPSK	30	0.0586
2	QPSK	50	0.0977
3	QPSK	78	0.1523
4	QPSK	120	0.2344
5	QPSK	193	0.3770
6	QPSK	308	0.6016
7	QPSK	449	0.8770
8	QPSK	602	1.1758
9	16QAM	378	1.4766
10	16QAM	490	1.9141
11	16QAM	616	2.4063
12	64QAM	466	2.7305
13	64QAM	567	3.3223
14	64QAM	666	3.9023
15	64QAM	772	4.5234

Apéndice D. Tablas CQI para LTE y 5G

Tabla D.2: Tabla para la elección de parámetros de acuerdo a CQI para LTE y 5G por defecto, extraída de [23] Tabla 5.2.2.1-2

CQI Index	modulation	code rate X 1024	efficiency
0	out of range		
1	QPSK	78	0.1523
2	QPSK	120	0.2344
3	QPSK	193	0.3770
4	QPSK	308	0.6016
5	QPSK	449	0.8770
6	QPSK	602	1.1758
7	16QAM	378	1.4766
8	16QAM	490	1.9141
9	16QAM	616	2.4063
10	64QAM	466	2.7305
11	64QAM	567	3.3223
12	64QAM	666	3.9023
13	64QAM	772	4.5234
14	64QAM	873	5.1152
15	64QAM	948	5.5547

Tabla D.3: Tabla para la elección de parámetros de acuerdo a CQI para LTE y 5G por configurable, notar que incluye esquemas 256QAM. Extraída de [23] Tabla 5.2.2.1-3

CQI Index	modulation	code rate X 1024	efficiency
0	out of range		
1	QPSK	78	0.1523
2	QPSK	193	0.3770
3	QPSK	449	0.8770
4	16QAM	378	1.4766
5	16QAM	490	1.9141
6	16QAM	616	2.4063
7	64QAM	466	2.7305
8	64QAM	567	3.3223
9	64QAM	666	3.9023
10	64QAM	772	4.5234
11	64QAM	873	5.1152
12	256QAM	711	5.5547
13	256QAM	797	6.2266
14	256QAM	885	6.9141
15	256QAM	948	7.4063

Apéndice E

Uso del Framework en simulador Vienna

Para solicitar acceso al repositorio que contiene tanto el Framework como el simulador debe enviar un correo a alguno de los autores del presente proyecto explicando el uso que se le dará al mismo (alexis.muzante@fing.edu.uy, martin.rizzo@fing.edu.uy, enzo.davyt@fing.edu.uy). Para tener acceso al Framework y simulador debe tener licencia de uso del Vienna 5G SLS, el Instituto de Ingeniería Eléctrica posee una licencia, en caso de no pertenecer al instituto debe adjuntarse al correo la documentación que acredite la posesión de la licencia.

Este apéndice tiene como objetivo mostrar como utilizar el Framework implementado sobre el simulador Vienna 5G SLS.

Como primer paso se recomienda definir qué métrica se desea utilizar teniendo en cuenta los parámetros y operaciones disponibles. Luego a partir de lo definido se debe pensar de qué forma agrupar las capas con sus respectivas prefuncions .

Dentro de la carpeta principal se encuentra el archivo “generate_scheduler.m”, dicho archivo contiene la generación de un scheduler utilizando el algoritmo, se recomienda usarlo como plantilla para generar otros. Además se incluye el escenario “NR_framework.m” (en la carpeta “+scenarios”) como ejemplo de escenario que utiliza el Framework y “launcher_NR_framework.m” (en la carpeta “+launcher-Files”) como archivo de ejecución de una simulación utilizando el escenario antes nombrado.

El procedimiento para la creación de un scheduler en el Framework se reduce a los siguientes pasos :

1. Nos paramos en la carpeta principal del simulador. Los archivos “framework_scheduler.m” y “layer.m” se encuentran en dicha carpeta, por lo que se debe estar en ella para utilizar los objetos definidos en ambos archivos.
2. Creamos las capas correspondientes utilizando el objeto “layer” que se define de la siguiente forma:

```
layer ( type_layer , matrix , prefuncions )
```

Recordando que el type_layer debe ser “S” en caso de ser combinación lineal y “P” en caso de ser productoria. “matrix” es la matriz correspondiente a la

Apéndice E. Uso del Framework en simulador Vienna

capa y prefunctions debe ser un vector de tamaño la cantidad de entradas, conteniendo alguna de las opciones en cada elemento, las opciones disponibles son: “exp”, “log”, “cos”, “sin” e “id”.

3. Construimos el objeto “framework_scheduler” de la siguiente forma

```
fs=framework_scheduler(layers, tiebreak_type,
                        parameters_selection)
```

En “layers” debe colocarse un vector que contenga las capas ordenadas de la primera a la última. En “tiebreak_type” debe introducirse una de las opciones de desempate (“first”, “last” o “random”). Por último en “parameter selection” se colocará un vector binario que indica cuales serán los parámetros de entrada del simulador. Los parámetros disponibles ordenados son [“CQI”, “mean”, “queue_length”], se indicará con 1 los deseados y con 0 los descartados (la primera capa los tomará ordenados en su entrada por lo que se deben construir las matrices teniendo esto en cuenta). Al construir el objeto se validará que los ajustes sean adecuados, en caso de haber una incongruencia (por ejemplo elementos con diferente dimensión de la requerida, opciones no disponibles, capas cuyas matrices no tienen tamaño acordes) el Framework devuelve un error con un texto descriptivo indicando el origen y como solucionarlo.

4. Por último se debe guardar la configuración en un archivo por lo que se llama al método “save_to_file”. Debe llamarse la función pasando como parámetro el string del nombre deseado para el archivo.

```
fs.save_to_file("scheduler-name");
```

Se guardará el archivo en la carpeta “framework_scheduler_files” con el nombre deseado y extensión “.mat” .

5. En el escenario debe especificarse el scheduler “framework” e indicar el archivo a utilizar en la propiedad “filename”. Esto se reduce a agregar las siguientes dos líneas de código al escenario:

```
params.schedulerParameters.type = parameters.setting
    .SchedulerType.framework;
params.schedulerParameters.filename = "scheduler-
    name.mat";
```

6. Simular con el escenario modificado.

Se incluye además el archivo “gráficos.m” si indicamos la radiobase con la variable “bs” y corremos el programa hará algunos gráficos que resultan de interés para analizar schedulers.

Referencias

- [1] 3rd Generation Partnership Project. 3GPP. <https://www.3gpp.org>. Online; accessed 31 January 2020.
- [2] Eugenio Rey. *Telecomunicaciones Móviles. Capítulo 8 Redes Celulares, sección 2 Descripción*, pag 85. Marcombo, Barcelona, Cataluña, 1998.
- [3] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 3 - Spectrum for 5G. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages 27 – 37. Academic Press, 2018.
- [4] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 5 - NR Overview. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages 57 – 71. Academic Press, 2018.
- [5] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck. Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Communications Surveys Tutorials*, 20(3):2429–2453, 2018.
- [6] Peter Rost, Christian Mannweiler, Diomidis S Michalopoulos, Cinzia Sartori, Vincenzo Sciancalepore, Nishanth Sastry, Oliver Holland, Shreya Tayade, Bin Han, Dario Bega, et al. Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks. *IEEE Communications magazine*, 55(5):72–79, 2017.
- [7] 3GPP; technical specification 5g; nr; nr and ng-ran overall description; stage-2 (3gpp ts 38.300 version 16.3.0 release 16). Standard, ETSI 3rd Generation Partnership Project (3GPP), Nov 2020.
- [8] K. I. Pedersen, G. Pocovi, J. Steiner, and S. R. Khosravirad. Punctured Scheduling for Critical Low Latency Data on a Shared Channel with Mobile Broadband. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–6, 2017.
- [9] Y. Kim, J. Bae, J. Lim, E. Park, J. Baek, S. I. Han, C. Chu, and Y. Han. 5G K-Simulator: 5G System Simulator for Performance Evaluation. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–2, 2018.

Referencias

- [10] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi. End-to-End Simulation of 5G mmWave Networks. *IEEE Communications Surveys Tutorials*, 20(3):2237–2263, 2018.
- [11] F. Kaltenberger, G. d. Souza, R. Knopp, and H. Wang. The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap. In *WSA 2019; 23rd International ITG Workshop on Smart Antennas*, pages 1–5, 2019.
- [12] S. M. A. Zaidi, M. Manalastas, H. Farooq, and A. Imran. SyntheticNET: A 3GPP Compliant Simulator for AI Enabled 5G and Beyond. *IEEE Access*, 8:82938–82950, 2020.
- [13] Martin Müller, Fjolla Ademaj, Thomas Dittrich, Agnes Fastenbauer, Blanca 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, 12 2018.
- [14] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS 21.101 version 8.4.0 Release 8). Standard, ETSI 3rd Generation Partnership Project (3GPP), mar 2012.
- [15] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS 21.101 version 9.2.0 Release 9). Standard, ETSI 3rd Generation Partnership Project (3GPP), mar 2012.
- [16] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS 21.101 version 10.4.0 Release 10). Standard, ETSI 3rd Generation Partnership Project (3GPP), jul 2014.
- [17] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 4 - LTE — An Overview. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages 39 – 55. Academic Press, 2018.
- [18] E. Bodanese. A Brief Introduction to Heterogeneous Networks (HetNets) and its Challenges. In *IET International Conference on Communication Technology and Application (ICCTA 2011)*, pages 605–609, 2011.
- [19] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS 21.101 version 11.2.0 Release 11). Standard, ETSI 3rd Generation Partnership Project (3GPP), abr 2016.
- [20] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS

- 21.101 version 12.0.2 Release 12). Standard, ETSI 3rd Generation Partnership Project (3GPP), ene 2017.
- [21] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS 21.101 version 13.0.0 Release 13). Standard, ETSI 3rd Generation Partnership Project (3GPP), ene 2017.
- [22] Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP System (3GPP TS 21.101 version 14.1.0 Release 14). Standard, ETSI 3rd Generation Partnership Project (3GPP), jul 2017.
- [23] 3GPP; Technical Specification Group Radio Access Network; NR; Physical Layer Procedures for Data (3GPP TS 38.214 V15.3.0 2018-09 Release 15). Standard, ETSI 3rd Generation Partnership Project (3GPP), sep 2018.
- [24] ITU. Working Party 5D (WP 5D) - IMT Systems. <https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/Pages/default.aspx>. Online; accessed 7 March 2020.
- [25] ITU. Recomendación UIT-R M.2083-0. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-S.pdf. Online; accessed 11 February 2020.
- [26] Gerhard Fettweis, Holger Boche, Thomas Wiegand, Erich Zielinski, Hans Schotten, Peter Merz, Sandra Hirche, Andreas Festag, Walter Häffner, Michael Meyer, Ekehard Steinbach, Rolf Kraemer, Ralf Steinmetz, Frank Hofmann, Peter Eisert, Reinhard Scholl, Frank Ellinger, Erik Weiß, and Ines Riedel. The Tactile Internet ITU-T Technology Watch Report. Technical report, ITU, 08 2014.
- [27] Ahmet Yazar and Huseyin Arslan. Flexible Multi-Numerology Systems for 5G New Radio. *Journal of Mobile Multimedia*, 14:367–394, 01 2018.
- [28] K. Pedersen, G. Pocovi, J. Steiner, and A. Maeder. Agile 5G Scheduler for Improved E2E Performance and Flexibility for Different Network Implementations. *IEEE Communications Magazine*, 56(3):210–217, 2018.
- [29] Ing. Bruno Benedetti. Instituto de Ingeniería Eléctrica. Universidad de la República. Montevideo, Uruguay. 5G NR, Red de Acceso. https://eva.fing.edu.uy/pluginfile.php/256091/mod_resource/content/0/5G-NR_RdA-2019_v1.pdf. Online; accessed 18 Jun 2020.
- [30] Technical Specification Group Radio Access Network, Study on New Radio Access Technology Physical Layer Aspects (3GPP TS 38.802 version 14.2.0 Release 14). Standard, 3rd Generation Partnership Project (3GPP), Sep 2017.

Referencias

- [31] K. Pedersen, G. Pocovi, J. Steiner, and A. Maeder. Agile 5G Scheduler for Improved E2E Performance and Flexibility for Different Network Implementations. *IEEE Communications Magazine*, 56(3):210–217, 2018.
- [32] X. Lin, J. Li, R. Baldemair, J. T. Cheng, S. Parkvall, D. C. Larsson, H. Koorapaty, M. Frenne, S. Falahati, A. Grovlen, and K. Werner. 5G New Radio: Unveiling the Essentials of the Next Generation Wireless Access Technology. *IEEE Communications Standards Magazine*, 3(3):30–37, 2019.
- [33] Kazuki Takeda, Huilin Xu, Taehyoung Kim, Karol Schober, and Xingqin Lin. Understanding the Heart of the 5G Air Interface: An Overview of Physical Downlink Control Channel for 5G New Radio (NR), 10 2019.
- [34] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno. Flexible Resource Block Allocation to Multiple Slices for Radio Access Network Slicing Using Deep Reinforcement Learning. *IEEE Access*, 8:68183–68198, 2020.
- [35] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 12 - Beam Management. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages 241 – 249. Academic Press, 2018.
- [36] Front-matter. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages i – iii. Academic Press, 2018.
- [37] Roy Gardner. *Games for Business and Economics*. Wiley, 2003.
- [38] IEEE Lome G. Mason Christos Douligeris Ravi Mazumdar, Member. Fairness in Network Optimal Flow Control: Optimality of Product Forms. *IEEE Transactions on Communications*, 39(5), 1991.
- [39] DKH Tan FP Kelly, AK Maulloo. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *The Operational Research Society*, 49(3):237–252, 1998.
- [40] A. Chodorek, R. R. Chodorek, and A. Krempa. An analysis of elastic and inelastic traffic in shared link. In *2008 Conference on Human System Interactions*, pages 873–878, 2008.
- [41] Scott Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communications*, 13, 1995.
- [42] Dimitri P Bertsekas, Robert G Gallager, and Pierre Humblet. *Data Networks*, volume 2. Prentice-Hall International New Jersey, 1992.
- [43] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. A Quantitative Measure of Fairness and Discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984.

- [44] Ritesh Madan, Stephen P Boyd, and Sanjay Lall. Fast Algorithms for Resource Allocation in Wireless Cellular Networks. *IEEE/ACM Transactions on networking*, 18(3):973–984, 2009.
- [45] Guowang Miao, J. Zander, K. W. Sung, and S. B. Slimane. Fundamentals of Mobile Data Networks. 2016.
- [46] Ing. Claudina Rattaro. Instituto de Ingeniería Eléctrica. Universidad de la República. Montevideo, Uruguay. 5G Simulators. https://www.fing.edu.uy/owncloud/index.php/apps/files/?dir=/FMV_AI45G/Seminario/sesion6&fileid=5657038#pdfviewer. Online; accessed 31 Aug 2020.
- [47] Markus Rupp, Stefan Schwarz, and Martin Taranetz. *The Vienna LTE-Advanced Simulators: Up and Downlink, Link and System Level Simulation*. Signals and Communication Technology. Springer Singapore, 1 edition, 2016.
- [48] 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.
- [49] Ing. Bruno Benedetti. Instituto de Ingeniería Eléctrica. Universidad de la República. Montevideo, Uruguay. LTE - Long Term Evolution. https://eva.fing.edu.uy/pluginfile.php/255858/mod_resource/content/0/LTE_RdA-2019_v1.pdf. Online; accessed 9 Sep 2020.
- [50] 5G;NR; User Equipment (UE) Radio Access Capabilities. Section 4.1.2 Supported Max Data Rate. Standard, ETSI 3rd Generation Partnership Project (3GPP), oct 2018.
- [51] D. Avidor, S. Mukherjee, J. Ling, and C. Papadias. On Some Properties of the Proportional Fair Scheduling Policy. *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754), Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on, Personal, indoor and mobile radio communications*, 2:853, 2004.
- [52] A. Buyukoglu, M. Izzet Saglam, A. Kavas, and M. Kartal. An Efficient Throughput Averaging Method for Proportional Fair Algorithm Used in Mobile Networks. *2016 Advances in Wireless and Optical Communications (RTUWO), Advances in Wireless and Optical Communications (RTUWO), 2016*, pages 161 – 166, 2016.
- [53] Sun Zhishui, Yin Changchuan, and Yue Guangxin. Reduced-Complexity Proportional Fair Scheduling for OFDMA Systems. *2006 International Conference on Communications, Circuits and Systems, Communications, Circuits*

Referencias

- and Systems Proceedings, 2006 International Conference on*, 2:1221 – 1225, 2006.
- [54] S.O. (1) Aramide, B. (1) Barakat, Y. (1) Wang, S. (1) Keates, and K. (2) Arshad. Generalized Proportional Fair (GPF) Scheduler for LTE-A. In *2017 9th Computer Science and Electronic Engineering Conference, CEEC 2017 - Proceedings*, number 2017 9th Computer Science and Electronic Engineering Conference, CEEC 2017 - Proceedings, pages 128–132, (1)Department of Engineering Science, Faculty of Engineering and Science, University of Greenwich, 2017.
- [55] Share Tech Note. 5G/NR - Resource Allocation Type. https://www.sharetechnote.com/html/5G/5G_ResourceAllocationType.html. Online; accessed 24 Nov 2020.
- [56] Ee Mae Ang, KuokKwee Wee, Ying Han Pang, and Keat Keong Phang. A Performance Analysis on Packet Scheduling Schemes Based on an Exponential Rule for Real-Time Traffic in LTE. *EURASIP Journal on Wireless Communications & Networking*, 2015(1):1 – 12, 2015.
- [57] European Telecommunications Standards Institute. ETSI. <https://www.etsi.org>. Online; accessed 31 January 2020.
- [58] Benigno Rodríguez. *Differential STBC for OFDM based Wireless Systems*. 01 2007.
- [59] 3GPP TS 36.321; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) Protocol Specification (Release 13). Standard, ETSI 3rd Generation Partnership Project (3GPP), jun 2016.
- [60] Erik Dahlman, Stefan Parkvall, and Johan Sköld. Chapter 2 - 5G Standardization. In Erik Dahlman, Stefan Parkvall, and Johan Sköld, editors, *5G NR: the Next Generation Wireless Access Technology*, pages 7 – 25. Academic Press, 2018.
- [61] Unidad Reguladora de Servicios de Comunicaciones. URSEC: Cometidos. <https://www.gub.uy/unidad-reguladora-servicios-comunicaciones/institucional/cometidos>. Online; accessed 6 March 2020.

Glosario de Siglas

3GPP	3rd Generation Partnership Project
5G NR	5th Generation New Radio
ACK	ACKnowledge
AL	Aggregation Level
AMPS	Advanced Mobile Phone System
ARIB	Association of Radio Industries and Businesses
ATIS	Alliance for Telecommunications Industry Solutions
BestCQI	Best Channel Quality Indicator
BLER	Block Error Ratio
BPSK	Binary Phase Shift Keying
BR	Best Resources
BS	Base Station
BTS	Base Transceiver Station
BWP	Bandwidth Part
C-MTC	Critical Machine Type Communication
CA	Carrier Aggregation
CC	Carrier Component
CCEs	Control Channel Elements
CCSA	China Communications Standards Association
CDMA	Code Division Multiple Access
CDMA2000	Code Division Multiple Access ITU IMT-2000
CEPT	European Conference of Postal and Telecommunications Administration
CITEL	Inter-American Telecommunication Commission
CoMP	Coordinated Multipoint
CORESET	Control Resource Set
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CSI	Channel State Information
CSI-RS	Channel-State Information Reference Signals
D2D	Device-to-Device
DAS	Distributed Antenna System
DCI	Downlink Control Information
DFT	Discrete Fourier Transform
DL	DownLink

Apéndice E. Glosario de Siglas

DMRS	Demodulation Reference Signal
ECC	Electronic Communications Committee
ECDF	Empirical Cumulative Distribution
eMBB	Enhanced Mobile Broadband
EPC	Evolved Packet Core
EPDCCH	Enhanced Physical Downlink Control Channel
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FIFO	First In First Out
gNB	gNodeB
GPRS/EDGE	General Packet Radio Service - Enhanced Data Rates for GSM Evolution
GSM	Global System for Mobile communications
GSMA	GSM Association
HARQ	Hybrid Automatic Repeat-Request
HeU	Highest eMBB User
HSPA	High-Speed Packet Access
HTTP	Hypertext Transfer Protocol
IA	Inteligencia Artificial
ID	IDentification
IMT	International Mobile Telecommunications
IMT-2020	International Mobile Telecommunications-2020
IoT	Internet of Things
IP	Internet Protocol
ITU	International Telecommunication Union
ITU-R	ITU - Radiocommunications
LAA	License-Assisted Access
LDPC	Low Density Parity Check
LeU	Lowest eMBB User
LLS	Link Level Simulator
LPM	Link Performance Model
LQM	Link Quality Monitoring
LQM	Link Quality Model
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
MCS	Modulation Coding Scheme
MF	Macroscopic Fading
MIMO	Multiple Input, Multiple Output
mMTC	Massive Machine Type Communications
MTC	Machine Type Communication
NGMN	Next Generation Mobile Networks
NLS	Network Level Simulator
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
PBCH	Physical Broadcast Channel

PCI	Physical layer Cell ID
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared Channel
PF	Proportional Fair
PMI	Precoding Matrix Indicator
POO	Programación Orientada a Objetos
PPP	Poisson Point Process
PRACH	Physical Random Access Channel
PRB	Physical Resource Block
PSS	Primary SS
PTRS	Phase-Tracking Reference Signals
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAM	Red de Acceso Móvil
RAN	Radio Access Network
RAT	Radio-Access Technology
RB	Resource Block
RE	Resource Element
REGs	Resource Element Groups
RF	Radio Frecuencia
RF1	Rango de Frecuencia 1
RF2	Rango de Frecuencia 2
RI	Rank Indicator
RLC	Radio Link Control
RoI	Region of Interest
RR	Round Robin
RRC	Radio Resource Control
RRH	Remote Radio Heads
SC-FDMA	Single Carrier Frequency Division Multiple Access
SDOs	Standards Developing Organization
SIM	Subscriber Identity Module
SINR	Signal to Interference plus Noise Ratio
SINR-BLER	Signal to Noise Ratio - Block Error Rate
SLS	System Level Simulator
SR	Scheduling Request
SRS	Sounding Reference Signals
SS	Synchronizaton Signals
SSS	Secondary Synchronizaton Signals
sTTI	short Transmission Time Interval
TB	Transport Block
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access

Apéndice E. Glosario de Siglas

TMA	Tower Mounted Amplifier
TS	Time Slot
TSDSI	Telecommunications Technology Development Society, India
TTC	Telecommunications Technology Committee
TTI	Transmission Time Interval
TU Wien	Universidad Tecnológica de Viennal
UCI	Uplink Control Information
UE	User Equipment
UL	UpLink
UMTS	Universal Mobile Telecommunications System
URRLC	Ultra Reliable Low Latency Communications
URSEC	Unidad Reguladora de Servicios de Comunicaciones
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VNF	Virtual Network Function
WCDMA	Wide Code Division Multiple Access
WiMAX	Worldwide Interoperability for Microwave Access
WP5D	Working Party 5D

Índice de tablas

2.1. Estructura de la numerología y correspondiente máximo ancho de banda para los canales en 5G [27].	12
5.1. Tabla resumen de escenario de simulación para validación de modelo de tráfico	51
6.1. Tabla resumen de simulación para validación de Proportional Fair implementado en simulador Vienna 5G SLS	62
7.1. Tabla resumen del escenario para la validación de las grillas de recursos	71
7.2. Tabla resumen del escenario para la validación de la asignación de recursos a usuarios URLLC	80
8.1. Tabla resumen de escenario de simulación para probar el Proportional Fair Queue	108
D.1. Tabla para la elección de parámetros de acuerdo a CQI para $BLER < 0,00001$, extraída de [23], Tabla 5.2.2.1-4	125
D.2. Tabla para la elección de parámetros de acuerdo a CQI para LTE y 5G por defecto, extraída de [23] Tabla 5.2.2.1-2	126
D.3. Tabla para la elección de parámetros de acuerdo a CQI para LTE y 5G por configurable, notar que incluye esquemas 256QAM. Extraída de [23] Tabla 5.2.2.1-3	126

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

2.1. Mejoras introducidas en las versiones de LTE [17].	8
2.2. Agregado de portadoras [17].	9
2.3. Servicios que ofrece 5G.	11
2.4. Estructura de un PRB para diferentes espaciamentos entre portadoras [29].	13
2.5. Estructura de la trama de 5G para diferentes numerologías y ejemplo de minislot [4].	14
2.6. Funcionamiento del punctured scheduler.	17
2.7. Beamforming: el dispositivo puede medir en un conjunto de RS (Reference Signals) correspondientes a diferentes haces de enlace descendente con el objetivo de refinar la transmisión [35].	19
3.1. Conjunto de enlaces de capacidad limitada que es asignada a los usuarios.	22
3.2. Utilidad para tráfico elástico y tráfico inelástico en función del ancho de banda. Figura extraída de [41].	23
3.3. Esquema del algoritmo Round Robin para un solo canal.	29
3.4. Esquema del algoritmo Round Robin para siete subcanales.	30
4.1. Bloques del simulador Vienna 5G SLS.	34
4.2. Bucle de simulación de un chunk sobre los time slots [13].	39
4.3. Bucle de simulación implementado en el Vienna 5G SLS.	40
4.4. Representación de la jerarquía de clases de los schedulers.	41
5.1. Gráfico de elementos de la red de simulación para validación de modelo de tráfico.	50
5.2. Gráfico de bits enviados por cada usuario en la radiobase utilizada para validar el modelo de tráfico.	50
5.3. Gráfico de recursos asignados por cada usuario en la radiobase utilizada para validar el modelo de tráfico.	52
5.4. Gráfico de recursos asignados, déficit y tráfico requerido por usuario 2 en la radiobase utilizada para validar el modelo de tráfico.	52
5.5. Gráfico de excedente de bits asignados y cantidad de bits por recurso del usuario 2 en la radiobase utilizada para validar el modelo de tráfico.	53

Índice de figuras

6.1.	Diagrama de flujo del scheduler Proportional Fair Generalizado, la variable u corresponde a la cantidad de usuarios que tienen bits pendientes para transmitir y nRBs el número de recursos libres. . .	58
6.2.	Gráfico que representa el CQI por usuario en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$	60
6.3.	Gráfico que representa la métrica por usuario (al comienzo de cada slot) en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$. . .	61
6.4.	Gráfico que representa la asignación de recursos por usuario en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$	61
6.5.	Gráfico que representa los bits alojados en los recursos por usuario en la radio base 2 de la simulación realizada para la validación del algoritmo de scheduler Proportional Fair con $\alpha = 1$ y $\beta = 1$	63
6.6.	Gráfico de barras de bits transmitidos por usuario en radiobase 5 en simulación de algoritmo Proportional Fair con $\alpha = 0$ y $\beta = 1$. .	64
6.7.	Gráfico de barras de recursos asignados por usuario en radiobase 5 en simulación de algoritmo Proportional Fair con $\alpha = 0$ y $\beta = 1$. .	64
6.8.	Gráfico de barras de recursos asignados por usuario en radiobase 2 en simulación de algoritmo Proportional Fair con $\alpha = 1$ y $\beta = 0$. .	65
6.9.	Gráfico del CQI promedio por usuario extraído del feedback en la radiobase 5 para la simulación con algoritmo Proportional Fair de $\alpha = 1$ y $\beta = 0$	66
6.10.	Gráfico de barras de recursos asignados por usuario en radiobase 5 en simulación de algoritmo Proportional Fair con $\alpha = 0$ y $\beta = 0$. .	67
7.1.	Representación de las grillas de recursos que maneja el simulador. (a) Grilla LTE, (b) Grilla NR, (c) Grilla NR con minislots.	70
7.2.	Evolución de la grilla de recursos correspondiente al primer slot de la simulación con grilla NR.	72
7.3.	Distribución espacial de BS y usuarios en la simulación con grilla NR.	73
7.4.	Distribución espacial de BS y usuarios en la simulación con grilla NR_minislot.	73
7.5.	Evolución de la grilla de recursos correspondiente al primer slot de la simulación con grilla NR_minislot.	74
7.6.	CQI promedio de los RB asignados a los usuarios en las simulaciones con grilla NR y NR_minislot.	75
7.7.	RB asignados a los usuarios en las simulaciones con grilla NR y con grilla NR_minislot.	75
7.8.	Bits asignados a los usuarios en las simulaciones con grilla NR y con grilla NR_minislot.	76
7.9.	Escenario para la validación del Proportional Fair con Puncturing funcionando con BR. Con rojo se representan las BS, con verde los usuarios URLLC y con azul los usuarios eMBB.	81

7.10. Asignación de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en BR. (a) Usuario 26. (b) Usuario 29. 82

7.11. Asignación de recursos al usuario 29 (URLLC) correspondientes al 1 minislot y al usuario 26 (URLLC) en el segundo minislot. 82

7.12. Asignación de recursos al usuario 26 (URLLC) correspondientes al quinto y sexto minislot. 83

7.13. CQI Promedio de los RB asignados a los usuarios en la validación del Proportional Fair con Puncturing funcionando en BR. 83

7.14. Escenario para la validación del Proportional Fair con Puncturing funcionando en HeU. Con rojo se representan las BS, con verde los usuarios URLLC y con azul los usuarios eMBB. 84

7.15. Asignación de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en HeU. (a) Usuario 23. (b) Usuario 27. 85

7.16. Asignación de recursos para el usuario 27 en la validación del Proportional Fair con Puncturing funcionando en HeU (a) Minislot 1 (b) Minislot 2. 85

7.17. Asignación de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en HeU (a) Minislot 3, usuarios 23 y 27 (b) Minislot 5, usuario 23. 86

7.18. CQI Promedio de los RB asignados a los usuarios en la validación del Proportional Fair con Puncturing funcionando en HeU. 86

7.19. Escenario para la validación del Proportional Fair con Puncturing funcionando en LeU. Con rojo se representan las BS, con verde los usuarios URLLC y con azul los usuarios eMBB. 87

7.20. Asignación de recursos para los usuarios URLLC (en el primer minislot) en la validación del Proportional Fair con Puncturing funcionando en LeU. (a) Usuario 21. (b) Usuario 25. 87

7.21. CQI Promedio de los RB asignados a los usuarios en la validación del Proportional Fair con Puncturing funcionando en LeU. 88

7.22. Asignación final de recursos para los usuarios URLLC en la validación del Proportional Fair con Puncturing funcionando en LeU. 88

8.1. Diagrama del sistema utilizado para el cálculo de la métrica. 93

8.2. Diagrama de objetos implementados para el Framework. 96

8.3. Diagrama de ejemplo del cálculo de la métrica. 97

8.4. Objetos que contienen los ajustes para generar el scheduler de ejemplo. 97

8.5. Diagrama de flujo para el Framework implementado. “u” representa la cantidad de usuarios cuya cola de datos no esta vacía y “nRBs” la cantidad de recursos restantes. 100

8.6. Bits enviados por usuario algoritmo justo implementado con el Framework. 102

8.7. Bits enviados por usuario algoritmo Proportional Fair Generalizado con $\alpha = 0$ y $\beta = 1$ 102

Índice de figuras

8.8. Recursos asignados por usuario utilizando el algoritmo best CQI implementado con el Framework.	102
8.9. Promedio del CQI por cada usuario utilizando el algoritmo best CQI implementado con el Framework.	103
8.10. Recursos asignados por usuario utilizando el algoritmo Proportional Fair implementado con el Framework.	103
8.11. Promedio del CQI por cada usuario utilizando el algoritmo Proportional Fair implementado con el Framework.	104
8.12. Posición de elementos de la red durante simulación de ejemplo para el Proportional Fair Queue.	106
8.13. Bits transmitidos por usuarios en primeros 50 slots del chunk 2 en la radiobase 2.	106
8.14. CQI por usuarios en primeros 50 slots del chunk 2 en la radiobase 2.	107
B.1. Evolución de LTE [17].	119
B.2. Tamaño del slot y PRB en downlink LTE [49].	119
C.1. Visión gráfica y simplificada de la relación entre las SDOs, cuerpos reguladores y administrativos y foros industriales [60].	123

Esta es la última página.
Compilado el lunes 1 febrero, 2021.
<http://iie.fing.edu.uy/>