ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Máster

DIVISIÓN DE FUNCIONES EN ARQUITECTURAS C-RAN
(Functional Split in C-RAN Architectures)

Para acceder al Titulo de

Máster Universitario en Ingeniería de Telecomunicación

Autor: Paula Sarasúa Zubimendi

OCTUBRE - 2017

Resumen

Las previsiones para los próximos años respecto a la evolución de las redes de comunicación actuales, pasan por un importante aumento de demanda de tráfico en las redes móviles, causando situaciones de congestión y de retardo en las mismas. Dicho incremento en la demanda de tráfico, se produciría por el aumento notable de nodos y por la consolidación de servicios de gran capacidad, tales como el vídeo streaming o los juegos online, que conviven con otros más básicos como la navegación web. En respuesta a estos problemas, el Third Generation Partnership Project (3GPP) propone, entre otras soluciones, el Cloud Radio Access Network (C-RAN) como posible arquitectura de red de acceso de los futuros sistemas 5G.

El principio básico de funcionamiento de esta nueva arquitectura consiste en llevar la inteligencia de los elementos de acceso de las redes de comunicación a la nube, basándose en tres grandes pilares: coordinación, centralización y virtualización de red. Además, surge como solución que satisface las nuevas necesidades de hoy en día por parte de los usuarios finales y de los operadores, reduciendo los costes en el despliegue de nuevos nodos en las redes de comunicación. Esta nueva filosofía no supondría ningún perjuicio para el usuario final, ya que el acceso se acercaría a la aplicación, y disminuiría la congestión y el retardo en el enlace de acceso.

En este trabajo se plantea un análisis de viabilidad de estas técnicas mediante la creación de una herramienta de simulación de acceso de redes $4G$. Haciendo uso de las posibilidades que ofrece la herramienta desarrollada y utilizando optimización basada en programación con restricciones, se han implementado los algoritmos correspondientes para calcular las posiciones óptimas de los controladores que gestionarán los nodos de la red y el nivel de división de funciones óptimo para cada elemento de acceso.

Índice

Índice de Figuras

Índice de Tablas

Lista de Códigos

Glosario

3GPP Third Generation Partnership Project. 2, 18, 21

4G Cuarta Generación. 9, 12, 21

5G Quinta Generación. I, 3, 4, 6, 7, 9, 10, 12, 17, 18, 21, 28, 58

AAS Active Antenna Systems ó Sistemas de antena activa. 10

AMC Adaptive Modulation and Coding. 19

BAB Branch and Bound. 50

BBU Band Base Unit. 18, 54, 59

BBUs Band Base Units. I, 1, 2, 15, 18, 27, 28, 30, 49

C-RAN Cloud Radio Access Network. 1–4, 18, 21, 28, 58, 59

CoMP Coordinated Multipoint. 10, 11, 14, 18, 21

COP Constraint Optimization Problems. II, 48

COTS Commercial off-the-shelf. 13

CSPs Constraint Satisfaction Problems (Problemas de satisfacción por restricciones). 44, 45

D2D Device-to-device. 18

DA Despliegue Aleatorio. 37

DFS Depth First Search. 50

DT Despliegue Teselado. 37

DUDe Downlink/Uplink Decoupling. 18, 21

FD-MIMO Full-Dimension MIMO. 10

GWNSyM Generic Wireless Network System Modeler. I, III, 21–23

IP Internet Protocol. 10

JRRM Joint Radio Resource Management. 14

LCV Least Constraining Value. 46

LDS Limited Discrepancy Search. 50

LTE Long Term Evolution. 1, 4, 6, 7, 9, 18

MIMO Multiple Imput - Multiple Output (múltiple entrada - múltiple salida). III, 7, 8, 19

MRV Minimum Remaining Values. 46

NFV Network Function Virtualization. 3, 9, 22

PaaS Platform Plataforma virtualizada como servicio. 6

PCDP Packet Data Convergence Protocol. 13–15

RAN Radio Access Network. 10, 13

RIC Rango intercuartílico. 40

ROHC Robust Header Compression. 15

RRH Remote Radio Head. 10, 11

SDN Software Define Network. 9, 22

XDSL any type of Digital Subscriber Line. 16

Capítulo 1

Introducción

Debido a las limitaciones que pueden encontrarse en las redes actuales de comunicación, tales como: restricciones en potencia, retardos demasiado elevados debidos a la congestión de la red o la popularidad de nuevos servicios que requieren gran capacidad, aparece como alternativa una nueva arquitectura, el C-RAN.

Esta nueva filosofía consiste en llevar la inteligencia de los elementos de acceso de las redes de comunicación a la nube, y surge como solución factible para satisfacer las nuevas necesidades de hoy en día por parte de los usuarios finales y de los operadores, que reducirían los costes en el despliegue de nuevos nodos en las redes de comunicación.

Esta nueva arquitectura no supondría ningún perjuicio para el usuario final, todo lo contrario, ya que el acceso se acercaría a la aplicación, y disminuiría la congestión y el retardo en el enlace de acceso.

Debido a la situación planteada, se ha considerado interesante realizar una simulación del comportamiento que tendría en una red Long Term Evolution (LTE), al utilizar virtualización de red realizando división de funciones en una arquitectura de tipo cloud. Utilizando las tecnologías adecuadas en el despliegue se consigue que la experiencia de usuario respecto a la capacidad y al retardo no se vea perjudicada, a la vez que se reduce el coste para los operadores de red. Para ello, se ha implementado una herramienta mediante programación C++ con la capacidad de modelar el fronthaul de una red de comunicaciones en la que se despliegan distintos tamaños de elementos de acceso al medio y BBUs.

A partir de las funcionalidades que ofrece la herramienta, se podrá desplegar y analizar una red concreta, con el objetivo de analizar el impacto económico que supondría la utilización de una arquitectura de red basada en virtualización y división de funciones de red.

Por todo esto los objetivos principales del trabajo serán:

- Creación de una herramienta de simulación de redes LTE.

- Implementación de un algoritmo que permita encontrar la posición óptima de los controladores a partir de algoritmos heurísticos.
- Implementación de un algoritmo, haciendo uso de la librería Gecode, para encontrar la división de funciones óptima en una red concreta, haciendo uso de programación con restricciones.
- Aplicar todos los objetivos anteriores para obtener en un escenario concreto, una distribución de BBUs óptima y, además, la división de funciones de red óptima para cada elemento de acceso.

Para dar respuesta a estos objetivos se ha dividido el trabajo en los capítulos que se comentarán a continuación, siendo el primero de ellos la introducción que se plantea en estas líneas.

- Capítulo 2: Consta de un estudio previo teórico en el que se detallan los conceptos necesarios para comprender en que consiste la arquitectura C-RAN.
- Capítulo 3: En este tercer capítulo de la memoria se desgrana cada una de las partes en las que está dividida la herramienta que se ha implementado para acometer el análisis, además de realizarse una comparativa con otras herramientas de simulación que existen en la actualidad.
- Capítulo 4: El estudio que se realiza a partir de la herramienta, se ha dividido en dos partes diferenciadas. Una primera parte, se presenta en el Capítulo 4, donde se hará únicamente uso de esta herramienta modular, implementando un algoritmo, basado en la solución de greedy, y obteniendo el mejor posicionamiento de las unidades de banda base según las estaciones base que estén desplegadas en un escenario concreto y con ciertos parámetros de configuración.
- Capítulo 5: En este Capítulo se plasma la segunda parte del estudio, donde se hará uso de una combinación de esta herramienta de simulación con la utilización de programación con restricciones, mediante el uso de la librería Gecode. Se acometerá el análisis que indique cual es la división de funciones óptima en los elementos de acceso, a partir del algoritmo generado en la primera fase de estudio.
- Capítulo 6: Por último se obtendrán las conclusiones de los resultados obtenidos, analizando la efectividad de los algoritmos planteados y el coste óptimo encontrado utilizando la herramienta desarrollada y la programación con restricciones, así como las posibles líneas futuras que se pueden aplicar a este trabajo.

Capítulo 2

Arquitectura C-RAN para 5G

En este apartado del trabajo se pretende explicar porqué surge como posibilidad de arquitectura de acceso al medio el C-RAN y cuál sería su estructura y composición, además de comentar las ventajas y desventajas que conllevaría [2] y [3].

A causa de la evolución de la tecnología y su gran repercusión en nuestro día a día, surge la necesidad de crear una nueva arquitectura de acceso de red para los futuros sistemas 5G, pudiendo así, encontrar la posibilidad de hacer frente a los nuevos requisitos de capacidad de red, de cobertura y de experiencia de usuario que exige la sociedad actual. Al mismo tiempo, por parte del operador de red, se plantea como objetivo principal para esta nueva arquitectura, reducir los costes y el tiempo de puesta en marcha de los servicios.

Para lograr lo que se plantea con anterioridad, es necesario un esfuerzo por parte de los operadores en todos los aspectos que engloban la red, ya que implica varios estándares, bandas frecuenciales, y diferentes soluciones de transporte de red. Por todo esto, la infraestructura que desarrollen debe de ser flexible y, al mismo tiempo, soportar y tener la capacidad de gestionar las redes de acceso radio heterogéneas que surjan. Actualmente, ha emergido la Network Function Virtualization (NFV) como un nuevo enfoque para conseguir esa flexibilidad de red requerida.

Al contrario de lo que sucedía hasta el momento, tratar de distribuir la red al máximo, la futura C-RAN, pretende realizar una combinación de técnicas de virtualización, con centralización de red y coordinación de los nodos.

Se van a presentar, a continuación, diferentes Secciones donde se documenta como está siendo la evolución de las redes actuales en dirección a la arquitectura C-RAN, sus impulsores, sus desafíos y las posibilidades que proporcionaría.

2.1. Evolución de la arquitectura de red de acceso al medio

La utilización masiva de teléfonos móviles conectados a la red requiere unas condiciones exigentes de cobertura, velocidad y latencia, como se explicaba anteriormente. Para hacer viables estos requisitos se está llevando a cabo una evolución del actual LTE y un planteamiento del futuro 5G (Figura 2.1), que traerá como resultados:

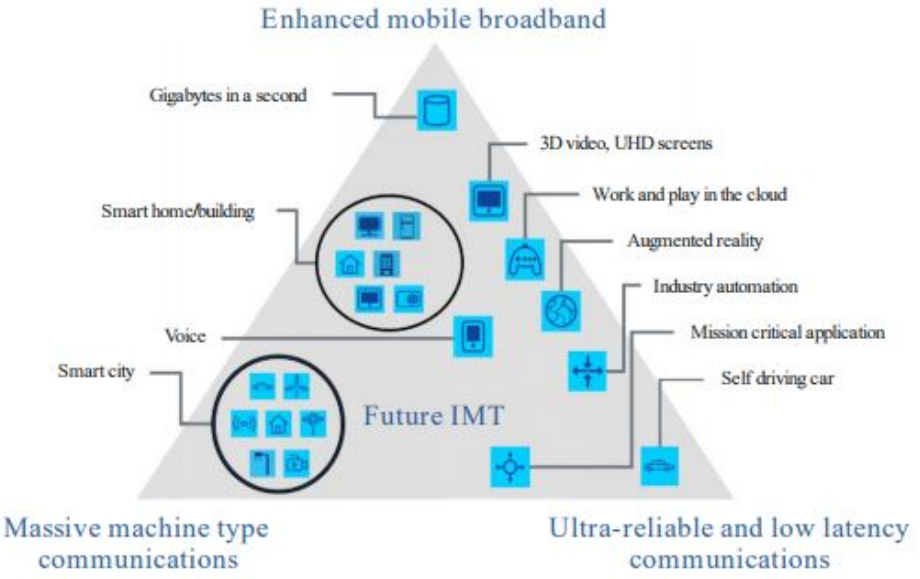


Figura 2.1: Arquitectura RAN en 5G [1]

- Una experiencia de usuario en bajada de 10-100Mbps garantizada en cualquier parte, y de 1-10Gbps localmente localizada.
- Una latencia garantizada, en las condiciones anteriores de bajada, de 1ms como máximo.

Para cubrir estas características es necesario moverse a otra parte del espectro, ocupando bandas más altas, por encima de los 4GHz. Es importante reseñar que se prevee su utilización en combinación con las bandas ya existentes.

Por otro lado, también será necesario desarrollar nuevos modelos de negocio, para soportar nuevos servicios y desarrollar el hardware y software necesario para la implementación de los mismos de una manera sostenible.

2.2. Impulsores de la arquitectura Cloud-RAN

En este apartado se van a explicar los tres grandes pilares en los que se va a basar la arquitectura C-RAN:

- Coordinación
- Centralización
- Virtualización de red

Estas tres características de una red serán las que sienten las bases de este trabajo. El objetivo conjunto de las tres es cumplir con los requisitos de capacidad y latencia que se comentaban con anterioridad.

2.2.1. Coordinación

Debido a la heterogeneidad de las futuras redes, de los diferentes tamaños de celda que van a coexistir, y las diferentes bandas espectrales que van a utilizar cada una de ellas, es una necesidad básica la coordinación entre las diferentes entidades que componen la red.

Dentro de lo que se podrían categorizar como mecanismos para lograr esa coordinación, se pueden encontrar:

- Gestión de movilidad, con el manejo de los traspasos
- Gestión de tráfico, balanceando la carga de las estaciones base
- Gestión de la interferencia, haciendo un control de la misma mediante coordinación de celdas potencialmente interferentes
- Recepción y transmisión conjunta
- Agregación de portadoras
- Conectividad dual

Con todo esto se concluye, que se gana en capacidad de red gracias a esta coordinación entre elementos, que por otro lado, requiere una comunicación directa de unas estaciones base con otras y tener conocimiento de todos los datos que manejan las demás, para saber en que momento es rentable coordinarse con unas, con otras, o simplemente apagarse si la interferencia que generaría esa estación base, supera las ventajas correspondientes a tenerla encendida.

2.2.2. Centralización

Como planteábamos en el apartado anterior, para poder coordinar las estaciones base, tienen que tener toda la información necesaria las unas de las otras. Por ello se pretende centralizar la gestión de las mismas por un ente único que las controle, y ellas simplemente ejecuten las órdenes que éste indique.

Centralizar el procesamiento de las estaciones base proporciona varias ventajas:

- Operacional: simplifica la gestión de red
- Hardware: según [2], la ganancia en multiplexado conseguida permitiría reducir de 10 a 100 celdas, dependiendo de la distribución de la carga y de los requisitos de servicio.
- Eficiencia espectral

Por otro lado, no hay que olvidar las desventajas que esto conlleva, ya que la complejidad computacional aumenta exponencialmente con la cantidad de elementos de acceso en una red.

2.2.3. Virtualización de red

La utilización de funciones de virtualización de red, es un recurso utilizado por los operadores para desplegar servicios en un corto plazo de tiempo con costes bajos. Para conseguir esto, hace falta superar los retos de diseño, integrando dispositivos con alta complejidad hardware en un mundo completamente interconectado basado en redes centralizadas.

Las funciones de virtualización de red nacen para el núcleo de la misma, pero tras su evolución, también se pueden aplicar al enlace radio. La idea es, ejecutar una plataforma software y hardware (que podría basarse en un entorno de Plataforma virtualizada como servicio (PaaS Platform)), con aplicaciones basadas en la nube, que posiblemente tengan aspectos críticos de latencia.

Esos aspectos críticos de latencia de las capas bajas (milisegundos, o incluso nanosegundos), van a limitar qué funciones pueden llevarse a la nube y cuáles no. Realizar un análisis para encontrar los aspectos de la funcionalidad del acceso al medio que podrían virtualizar, es uno de los principales objetivos que se plantean en este Trabajo de Fin de Máster, y proporcionará la capa y las tecnologías adecuadas a partir de la cual es interesante llevar esas funciones de las estaciones base a la nube, para un escenario concreto.

2.3. Desafíos de la red de acceso del futuro

A lo largo de esta Sección se van a presentar los principales desafíos del acceso radio del futuro, que podrían tener un gran impacto en el desarrollo de la arquitectura del 5G. La estandarización del 5G aún está por terminar de especificar, y podría dividirse en dos estándares:

- Una evolución natural del actual LTE, con principal objetivo de mejora de funcionalidad, a la vez que se asegura una compatibilidad hacia atrás con el LTE.

- Una nueva tecnología de acceso radio que se situaría en bandas de frecuencia diferentes a las de LTE. Esta tecnología se construiría en base a cumplir los requisitos del 5G, tales como nuevos casos de uso, como la disponibilidad de múltiples bandas frecuenciales y diferentes opciones de acceso.

A continuación se presentan las diferentes dificultades que se tienen que abordar en la tarea de implantar esta nueva arquitectura, y cómo se plantea superarlas.

2.3.1. Espectro para el 5G

El principal objetivo respecto al espectro de esta nueva tecnología de acceso radio, es pasar de tener agregaciones de portadoras de MHz a GHz. Para conseguir este objetivo, la única solución es tener la capacidad de subir en frecuencia, por encima de los 6GHz, esto es, ondas milimétricas.

Por todo esto, la solución pasa por encontrar el espectro adecuado en el rango 6-100GHz, que esté infrautilizado o que no se haya utilizado hasta el momento. Pero como toda solución que pasa por subir en frecuencia, surgen numerosos problemas, como las pérdidas por propagación, que aumentan considerablemente.

Finalmente, el objetivo que se busca respecto al espectro frecuencial pasa por:

- Tener una cobertura global en frecuencias por debajo de los 6GHz
- Tener cobertura parcial en bandas superiores a los 30GHz

2.3.2. Capa física y gestión de recursos radio

Debido al objetivo que se plantea en el apartado anterior, se busca cómo habilitar el uso de portadoras a altas frecuencias, mediante el uso de MIMO masivo y técnicas de conformado de haz (Figura 2.2).

Además, el uso de conformado de haz en transmisión y recepción proporciona una gran autonomía, reduciendo la posibilidad de caída del enlace radio a altas frecuencias. Por otro lado, estas técnicas suponen aumentar la complejidad de las entidades de red y la sobrecarga.

No hay que olvidar que también hay que posibilitar la armonización del estándar que propone la evolución del acceso radio LTE con esta nueva tecnología. Se ha probado conceptualmente que los esquemas de control del LTE son lo suficientemente robustos para poder realizar el control de los recursos radio en estas nuevas bandas de frecuencia.

Debido a esto, tanto la nueva tecnología como la evolución del LTE actual que planteábamos como las dos opciones de acceso radio para 5G, se conectarán al core de la red a través de un interfaz radio, como se plantea en el esquema de la Figura 2.3.

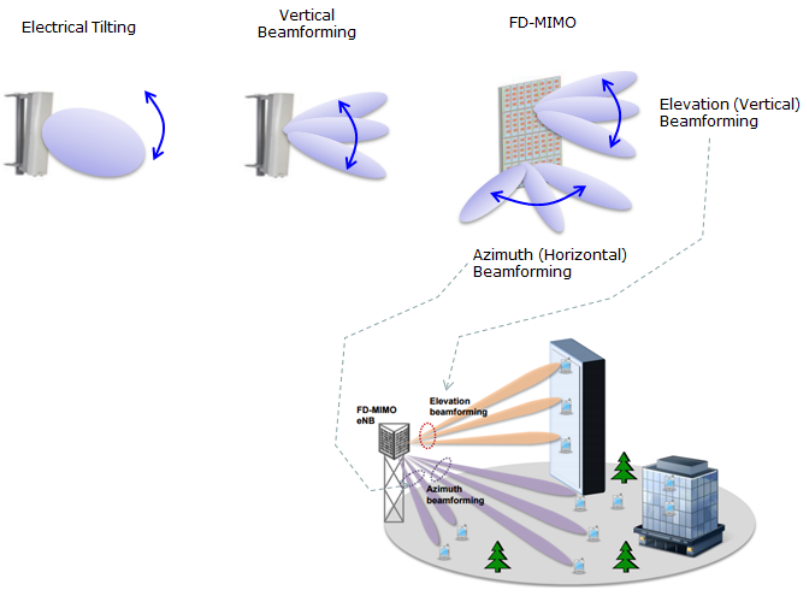


Figura 2.2: MIMO masivo y conformado de haz

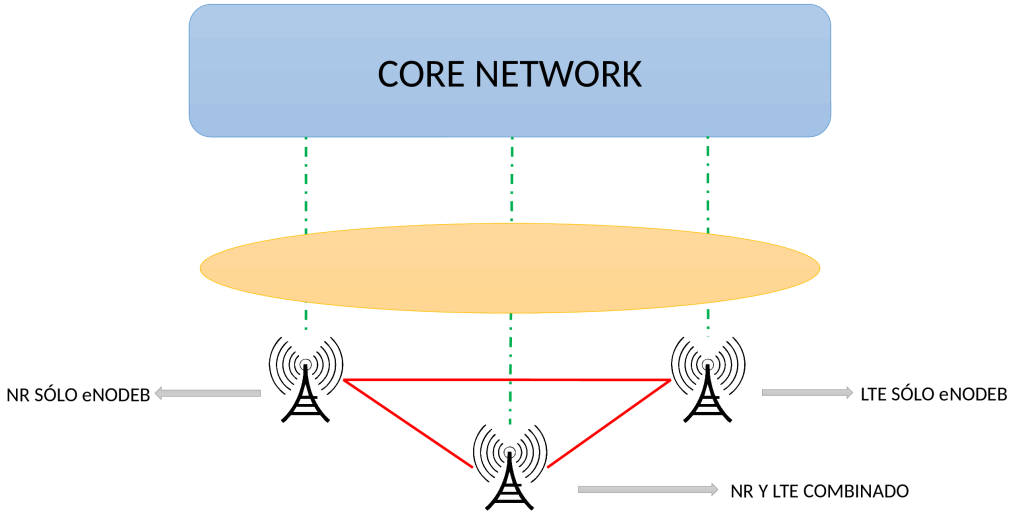


Figura 2.3: Conexión lógica de la arquitectura de LTE con el nuevo acceso radio

Además de este primer problema de armonización entre las bandas, la nueva tecnología de acceso radio tiene que tener una latencia muy baja. Eso hace necesaria la implementación de tiempos de intervalo cortos y flexibles y una nueva estructura de canal.

2.3.3. División de funciones de red

Debido a que del 5G se espera el soporte de una gran cantidad de servicios, con unos requisitos de ancho de banda y latencia estrictos sobre múltiples escenarios, se propone poder ser soportado a través de la división de funciones de red.

Este concepto hace que el usuario final, o el operador de red, vean la parte de red a la que está accediendo como una entidad lógica separada de las demás, cómo si fuese una red dedicada. Pero en realidad comparte la parte de procesado, transporte y enlace radio con otras redes que se implementen sobre la misma infraestructura (Figura 2.4).

Se llega a este concepto, mediante el uso de los principios de funcionamiento de NFV y Software Define Network (SDN) manteniendo separadas la gestión de recursos radio y la visión lógica de la red.

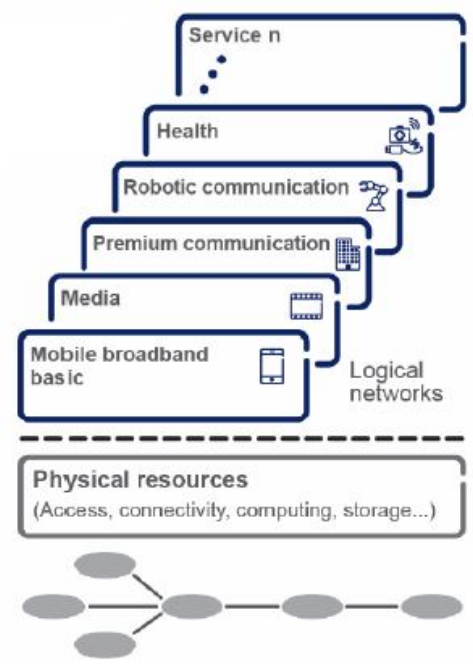


Figura 2.4: Principio de funcionamiento del network slicing

2.3.4. Evolución de LTE a 5G

Como se ha descrito anteriormente la evolución del LTE es una parte integrada en el 5G, por lo que alguno de los aspectos clave del 5G que se plantean a continuación, han sido ya estandarizados en la última release de la Cuarta Generación (4G), la actualización 14 [4]. En las siguientes líneas, se muestra una breve descripción de algunos de esos aspectos más relevantes:

- Active Antenna Systems ó Sistemas de antena activa (AAS): los sistemas de antena activa, son capaces de ajustar dinámicamente el patrón de radiación e introducir sectorización en los planos vertical y horizontal, además de beamforming. Estas técnicas, adquieren mucha importancia en escenarios muy densos, para mejorar la gestión de la interferencia.
- Beamforming y Full-Dimension MIMO (FD-MIMO): los sistemas de antena activa de los que se ha hablado con anterioridad son la base para construir sistemas de FD-MIMO. En este punto, es importante ver como aumenta la complejidad del sistema con el número de antenas. Así, dependiendo de esta complejidad, se podría elegir entre una arquitectura distribuida o centralizada. Se tratará el tema de las diferentes arquitecturas en apartados posteriores.
- Mejora de señal con la utilización de técnicas Coordinated Multipoint (CoMP): se implementarán procedimientos para distribuir la información de control dentro del conjunto de entidades que están coordinadas en CoMP, si la arquitectura es distribuida. En caso de ser centralizada, no haría falta, ya que la infraestructura al completo se gestionaría desde un único punto lógico.
- Mejoras con la utilización de conectividad dual: la conectividad dual jugará un rol muy importante en la arquitectura del 5G.

2.4. Diferentes posibilidades de arquitectura

A continuación se presentan las diferentes arquitecturas que se consideran para la arquitectura definitiva del 5G.

2.4.1. RAN distribuida

En una arquitectura de este tipo, el interfaz entre el Radio Access Network (RAN) y el núcleo de la red, se encuentra localizado en el Remote Radio Head (RRH). La mayoría de redes LTE en la actualidad se basan en este tipo de arquitectura, que proporciona un rápido y fácil despliegue y una conectividad basada en Internet Protocol (IP).

Este tipo de arquitectura proporciona funcionalidades tales como:

- Los traspasos no cortan la comunicación, se hacen manteniendo una continuidad en la conexión (Soft-handover)
- Soporta agregación de portadoras e implementa técnicas CoMP en recepción, lo que mejora la eficiencia
- Utilización de small cells como entidades en la red, haciendo posible la utilización de bandas frecuenciales más altas en el espectro

2.4.2. RAN centraliada

Del interés que había en distribuir la inteligencia de la red por parte de los operadores, se ha llegado con este nuevo concepto de RAN centralizada, a una búsqueda completamente opuesta. El interés por la centralización de la inteligencia de la red ha crecido, en lugares donde la densidad de nodos es muy grande, tales como: estadios o centro de ciudades donde se reúne gran cantidad de personas; y en escenarios donde se pueda llevar parte de las funcionalidades de los RRH a la nube.

En un despliegue de este tipo, todo el procesado del acceso radio, y las capas 1, 2 y 3 podría estar localizado en un nodo central, que sirve a las diferentes entidades que carecen de inteligencia, y sólo actúan en función de lo que ese nodo central comunica, como acciones que se tienen que tomar en cada momento.

Estos enlaces entre el nodo central y los elementos de red distribuidos, se pueden realizar sobre diferentes tecnologías: fibra óptica y radioenlaces. De las características de este tipo de enlaces en el fronthaul de la red se hablará a continuación, ya que los requisitos de esos enlaces se basan en distintas condiciones de capacidad y latencia, según el nivel de división de funciones de red impuesto por el controlador.

De estas características, que podrían limitar las comunicaciones, se va a justificar que la limitación fundamental para el enfoque que se propone, se sitúa en la latencia de la red, que acota la división de funciones de red, como se comentará en apartados sucesivos.

El potencial que tendría este tipo de arquitecturas es aún desconocido, porque no está implementado en ningún escenario real, pero se estima que se podría mejorar las tasas de bajada de un $40 - 70\,\%$ en escenarios superpoblados, simplemente habilitando la planificación coordinada por estos controladores de red.

Por otro lado, la mejora en el enlace de subida podría ser notablemente superior, aumentando la velocidad, entre 2 y 3 veces o incluso más, dependiendo de la potencia de la señal y los niveles de interferencia, que también serían gestionados por el controlador.

2.4.3. Combinar los despliegues de acceso distribuidos y centralizados

Dependiendo de las necesidades y de la disponibilidad de tecnologías, se podría llegar a la combinación de una arquitectura centralizada con una distribuida.

Si se posibilita la interconexión de las capas 1 y 2, las unidades de banda base podrán ser agregadas, habilitando coordinación sobre los despliegues centralizados y distribuidos. El usuario final se va a beneficiar siempre de los aspectos que proporciona la coordinación, como son CoMP o la agregación de portadoras, que les proporciona

un mayor ancho de banda, incluso cuando estén bajo la cobertura de cualquier tipo de celda en diferentes unidades banda base.

2.4.4. Aplicar técnicas de virtualización en el RAN

Todas las arquitecturas que se han presentado hasta el momento, han sido una solución viable para las redes 4G de la actualidad, pero desde el momento en que para el 5G se plantea una subida en frecuencia, con la aparición de las bandas de cobertura parcial, es necesaria una solución que pase por la virtualización de funciones de red.

Una arquitectura para el acceso radio basada en este tipo de técnicas tiene que lidiar tanto con el aspecto de la limitación de cobertura como con las diferentes capacidades soportadas en toda la banda disponible en el espectro para 5G. Debido a ello, la RAN virtualizada soporta:

- Coste eficiente, mediante el uso de compartición de procesado: controladores de red
- Flexibilidad en la capacidad hardware
- Trabajo entre capas: entre la capa de aplicación y toda la parte del acceso al medio
- Movilidad robusta

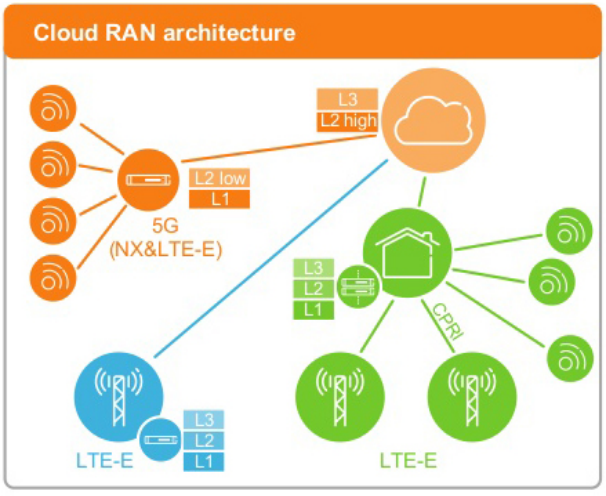


Figura 2.5: Partes de una RAN virtualizada

Uno de los beneficios más notables por el que se utiliza la virtualización de funciones de red, es que ofrece la separación o 'split' de las capas asíncronas de la pila de protocolos del acceso radio.

El principal beneficio que ofrece realizar este 'functional split' está relacionado con la importancia en la interrelación entre celdas de diferente tamaño, que trabajan a diferentes frecuencias y en distintos despliegues. Si se ofrece esta interrelación entre celdas, los recursos son utilizados más eficientemente y las bandas altas con cobertura parcial se pueden usar en mayor medida, manteniendo la conectividad fiable en las capas bajas.

Otro beneficio de virtualizar el acceso al medio, es que las funcionalidades que se separan de la unidad de banda base y se virtualizan, corren sobre un hardware genérico, obteniendo flexibilidad y permitiendo trabajar en conjunto con las funcionalidades del core.

Todas estas acciones comentadas, proporcionan tanto beneficios del lado de plano de usuario, como del lado del plano de control.

Respecto al plano de control, la virtualización permite a los operadores centralizar el plano de control, sin una exigencia muy alta de tasa binaria, acercando más la parte de la funcionalidad RAN a la capa de aplicación. Esta proximidad de ambas capas hacen posible una mejora del Commercial off-the-shelf (COTS), tanto en la flexibilidad y costes, como reduciendo el tiempo de puesta en marcha de los servicios. Todo esto hará que la latencia entre el RAN y el core de la red se reduzca.

Por otro lado, respecto al plano de usuario, los beneficios van hacia escenarios donde se posibilite la conectividad dual de los dispositivos. Con conectividad dual en un entorno distribuido, se produce un efecto de embotellamiento en la red, ya que el tráfico es encaminado ineficientemente. Este efecto se puede evitar, colocando el protocolo de encaminamiento en la capa de transporte, de tal modo que se mejora la latencia de la red.

La capa 2 en el plano del usuario, Packet Data Convergence Protocol (PCDP) es principalmente un protocolo, pero también tiene una parte importante de cifrado. Como un complemento para mejorar ese entorno de procesado genérico de paquetes, se podría optimizar el cifrado mediante aceleradores, para mejorar la latencia y la actuación en las bandas altas, de una manera energéticamente eficiente y con un coste aceptable.

2.5. Aspectos de diseño

En las secciones previas de este capítulo de la memoria se ha analizado, tanto la evolución de las redes de la actualidad que nos llevará a la nueva generación de comunicaciones móviles, como los retos tecnológicos que nos marcan esos requisitos y las diferentes soluciones con las que se podrían afrontar. Por otro lado, en esta nueva sección, se analiza cómo las arquitecturas propuestas, podrían ayudar a cumplir con esos requisitos identificados.

2.5.1. RAN functional splits

La centralización de las funciones de acceso radio, permite, como se ha dicho con anterioridad, una asignación inteligente de los recursos, en situaciones donde no todas las celdas van a demandar el 100 % de la capacidad de procesado al mismo tiempo. Por ello, es más fácil realizar Joint Radio Resource Management (JRRM) sin reestructurar el tráfico de datos entre los nodos.

A continuación, se presentan diferentes opciones donde poder realizar el 'split' de las funcionalidades [5] (Figura 2.6).

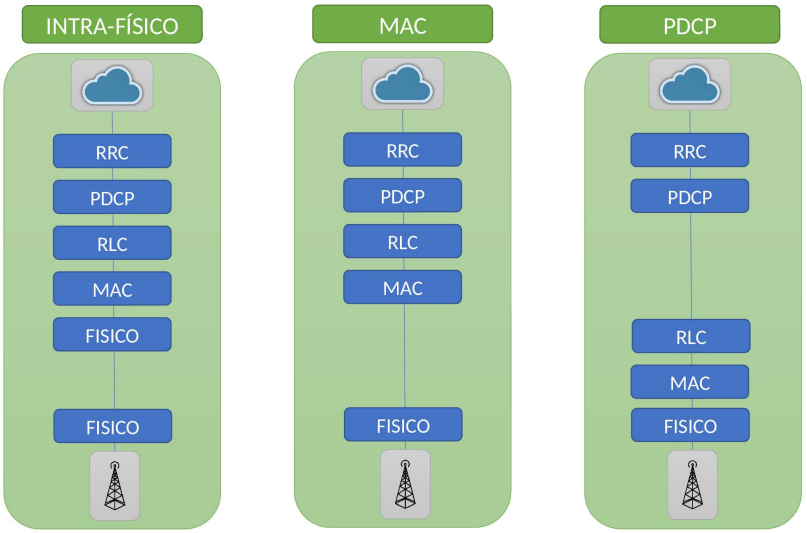


Figura 2.6: Posibilidades de realización del split

Una primera posibilidad, es colocar el split en la capa física (Figura 2.6, lado izquierdo), entre la precodificación y el mapeo de los recursos. Esta opción permite a técnicas, como el MIMO masivo y el CoMP ejecutarse sin una carga excesiva de datos entre los nodos. El tráfico en el fronthaul, puede explotar entonces la ganancia que le ofrecería la agregación de tráfico, de tal manera que se produciría un alivio de los requisitos en el transporte.

Otra posibilidad en la que realizar el split, podría ser entre las capas física y MAC (Figura 2.6, centro). Los beneficios en este caso, pasan por reducir la tasa binaria en el fronthaul, porque sólo los bits de los bloques de transporte pueden ser intercambiados de manera agregada, reduciendo así los requisitos de capacidad. Sin embargo, se podrían coordinar y centralizar menos funcionalidades, en comparación con el split intra físico.

Por último, un tercer split podría estar definido, en el punto más alto del plano de control dentro de la pila de protocolos, a nivel PCDP (Figura 2.6, lado derecho). Este split, permite la conectividad múltiple dividiendo el tráfico en diferentes flujos dirigido

a través de diferentes nodos de acceso. Con la centralización de PCDP, se obtienen además, mejoras en ganancia de multiplexado mediante la utilización de protocolos de compresión de cabeceras (Robust Header Compression (ROHC)).

Para simplificar el problema que se plantea en este trabajo, se van a reducir las posibilidades a la utilización de dos Splits, siendo estos los que se encuentran en el extremo izquierdo y el centro de la Figura 2.6, (nivel físico y MAC).

2.5.2. Aspectos de la red de transporte

La introducción de cualquiera de los splits que se han planteado, tendrá una repercusión en una red de transporte real, produciéndose:

- Nuevos interfaces físicos se deben definir en los puntos de split (BBUs)
- Para mantener la continuidad en las operaciones de la pila de protocolos en la parte del acceso al medio, los requisitos de la red de transporte deben ser más estrictos; tales como altos throughputs y bajas latencias, tal y como requieren los protocolos del interfaz aire y la estructura de trama.

2.6. Diferentes tecnologías utilizadas según el tamaño de celda y el nivel de división de funciones de red

Tal y como se observa en la Figura 2.7 [6], según el tipo de celda que se tenga se podrá utilizar una u otra tecnología:

- Macro celdas: pueden utilizar microondas, fibra óptica
- Micro celdas: pueden utilizar ondas milimétricas, fibra óptica y XDSL

A partir de la tecnología que se seleccione para el enlace de comunicación, se podrá acceder a uno u otro nivel de splitting, debido a las limitaciones de capacidad, alcance y latencia que tienen esas tecnologías, como se ve en la Tabla 2.1.

Tabla 2.1: Resumen de los parámetros de las posibles tecnologías del fronthaul

	Latencia	**Capacidad**	**Alcance**
Fibra Óptica	<2.5us	10Gbps	1km
Microondas	1ms	100Mbps-1Gbps	2km
Ondas milimétricas	1ms	500Mbps-2Gbps	1km
XDSL	15-60ms	10-100Mbps	2km

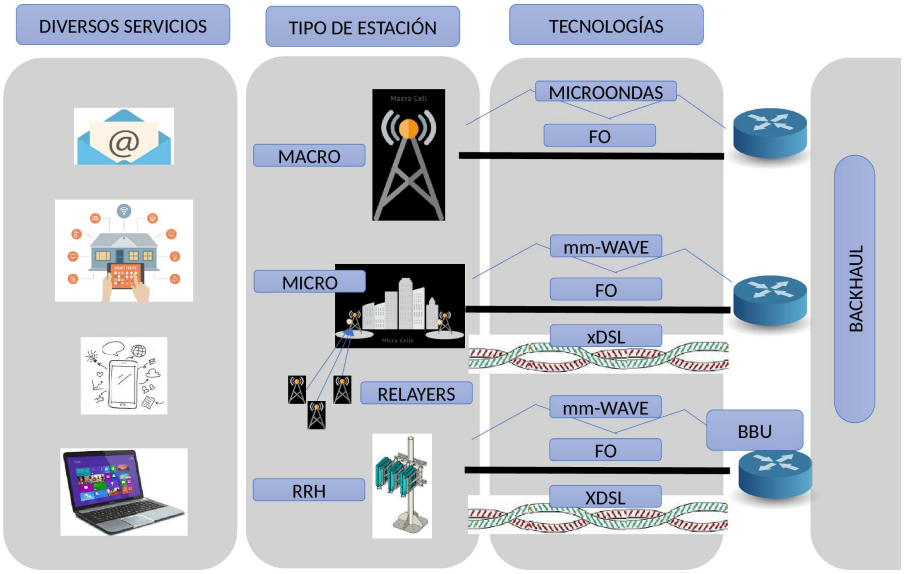


Figura 2.7: Tecnologías que se utilizan en función del tamaño de la celda

Respecto a las posibles tecnologías, entre las soluciones radio, existen dos opciones, dependiendo del tamaño del elemento de acceso. Para las small-cells, se utilizan ondas milimétricas, y para las macro celdas, microondas. Por otro lado, está la solución cableada, que en este caso se ha decidido que sólo se llevará a cabo por fibra óptica.

A partir de los siguientes apartados la tecnología any type of Digital Subscriber Line (XDSL) no se ha tenido en cuenta, por tener unas características insuficientes de capacidad y latencia para este tipo de comunicaciones.

Respecto a los costes de las diferentes tecnologías, se comentará el impacto que tiene sobre ellos la distancia, en los capítulos de resultados.

Dados estos parámetros, se tienen que seleccionar las tecnologías que van a poder funcionar a los diferentes niveles de splitting. Por ello es necesario conocer los requisitos de los niveles de división en funciones que se podrían plantear (Tabla 2.2) y que se explican en profundidad a partir de la Figura 2.6.

Tabla 2.2: Requisitos necesarios para los diferentes niveles de splitting

	Latencia	**Capacidad**
Lower Physical	<1ms	470Mbps
Upper Physical	3ms	100Mbps
MAC	3ms	70Mbps
PDCP	100ms	70Mbps

Una vez que se conocen los parámetros de las tecnologías y de los posibles splits, se van a reducir las posibilidades de este trabajo, a dos niveles de split: llevar todas las funcionalidades desde la capa física baja (split Físico) o llevar las funcionalidades desde la capa MAC (split MAC). Para resumir la situación se presenta la Tabla 2.3, donde se resumen las posibilidades que ofrecen las tecnologías según el split seleccionado.

Dependiendo del escenario concreto a analizar, se va a restringir directamente la fibra óptica para un split Físico y las ondas radio para un split MAC, o se seguirá directamente lo que indica la Tabla 2.3.

Tabla 2.3: Resumen de las posibles tecnologías con los respectivos Splits posibles

	Split Físico	**Split MAC**
Fibra Óptica	SI	SI
Microondas	NO	SI
Ondas milimétricas	NO	SI

En resumen, una vez tenidas en cuenta todas las consideraciones que supone la utilización de cada uno de los splits, no hay una única solución óptima a priori, ya que depende de las características del escenario a estudiar, por lo que la mejor solución pasa por tener una arquitectura 5G en la que se pueda adaptar la infraestructura de transporte según el caso, soportando, de esta manera, diferentes splits.

A lo largo de este trabajo se analizarán diferentes escenarios:

- En el primero de ellos, se definirá por configuración, según el tamaño del elemento de acceso, cuál es su nivel de splitting, y utilizando un algoritmo que ejecute una P-Mediana, se encontrará el despliegue cuasi-óptimo de los controladores.
- En el segundo escenario, se planteará una solución obtenida a través del algoritmo que selecciona la localización óptima de los controladores, y se analizará cual es la mejor distribución de splits que se podría plantear en ese caso para los elementos de acceso.

Capítulo 3

Entorno de simulación

En este tercer capítulo de la memoria se desgrana cada una de las partes en las que está dividida la herramienta [7] que se ha implementado para acometer el análisis, y se realiza una comparativa con otras herramientas de simulación de redes LTE que existen en la actualidad.

El entorno en el que se ha desarrollado esta herramienta es C++, teniendo como objetivo conseguir una plataforma de simulación de escenarios con redes LTE, mediante la adhesión de diferentes módulos a un núcleo común, proporcionándole funcionalidades y ciertos comportamientos adicionales según las técnicas empleadas, como podrían ser CoMP, Downlink/Uplink Decoupling (DUDe), Device-to-device (D2D) o una arquitectura C-RAN, las cuáles son técnicas que se proponen en las nuevas releases del 3GPP para el futuro 5G.

En el caso de este trabajo, el estudio se ha centrado en la implementación de algoritmos para estudiar el impacto del C-RAN en el futuro 5G, utilizando virtualización de red y división de funciones de red entre la nube y los elementos de acceso.

En concreto, se implementarán dos algoritmos diferentes:

- Un primer algoritmo basado en una solución greedy para resolver el problema de P-Mediana (Capítulo 4) y donde se pretende obtener, dentro de un despliegue de red, la posición más cercana al óptimo de las BBUs, teniendo como objetivo minimizar el coste (en distancia de enlace desplegada y en impacto económico según el nivel de división de funciones de red utilizado).
- Un segundo algoritmo (Capítulo 5) en el que se combina la posición óptima que se obtiene en el primer algoritmo con la utilización de programación con restricciones para obtener el split óptimo para cada Band Base Unit (BBU) y las estaciones base que están conectadas a esos controladores.

Una vez implementadas ambas fases del análisis, se estudiarán las métricas pertinentes para obtener resultados y analizar el comportamiento final de la red, que se presentarán en los Capítulos 4 y 5.

Previa a esa presentación de algoritmos y resultados es importante hablar de la herramienta desarrollada para la simulación de este tipo de escenarios, y la comparativa de la misma, con otras metodologías de simulación existentes en la actualidad.

3.1. Metodologías de simulación con distintos niveles de abstracción

Uno de los primeros problemas a la hora de comenzar un análisis concreto de un escenario es la elección de la herramienta a utilizar, en función del nivel de abstracción requerido. Las plataformas de simulación juegan un papel fundamental, debido a su versatilidad y coste. En este sentido, existen básicamente tres alternativas principales:

- Simuladores a nivel de enlace: su cometido se centra en la parte inalámbrica de la comunicación, es decir, el último salto hasta llegar a los usuarios finales. Este tipo de herramientas proporcionan la capacidad de analizar técnicas de enlace, estimación de canal, técnicas MIMO o soluciones relacionadas con el Adaptive Modulation and Coding (AMC). Dentro de este conjunto de simuladores se encuentra el Vienna LTE Simulator [8] como una de las soluciones más extendidas.
- Simuladores a nivel de sistema: los simuladores de este segundo tipo, permiten realizar un análisis con una mayor flexibilidad, a costa de realizar alguna simplificación, que, acarrea cierta pérdida en la precisión del análisis. La mayoría de los simuladores de este tipo se basan en desarrollos propietarios, en su mayoría, desarrollados en MATLAB. En algunas ocasiones se usan algunas de las pocas herramientas específicas disponibles, donde nuevamente destaca el simulador LTE Vienna [8].
- Simuladores de red: existen varias plataformas, aunque la más utilizada en los últimos tiempos es ns-3 [9], y su extensión LTE-EPC Network Simulator (LENA) [10], que se encuentra en constante evolución. En el caso de este tipo de simuladores, encuentran su mayor limitación en el tiempo necesario para acometer el análisis, que se produce por el alto grado de detalle de su implementación y modelos.

En la Tabla 3.1 se presenta una comparativa de los tres tipos de simuladores que se han comentado con anterioridad, basándose en sus características principales. En ella se presenta una clasificación de los parámetros, de forma que cuando el círculo está relleno, el significado es de buen comportamiento de la herramienta, mientras que los vacíos indican un rendimiento pobre en ese parámetro.

Una de las limitaciones que comparten las soluciones existentes es la dificultad para incluir o modelar nuevas técnicas y modelos de red, ya que su implementación suele ser bastante rígida, porque habitualmente está centrada en escenarios concretos.

Tabla 3.1: Análisis de características de alternativas de simulación para redes inalámbricas (LTE)

	Parámetro *Descripción de la característica de simulación que se necesita soportar*	**Simulación a nivel de enlace** *Modelado detallado de capas inferiores, lo que dificulta analizar escenarios con mas de un par fuente/destino*	**Simulación a nivel de sistema** *La mayoría de la literatura usa Matlab para realizar análisis. Vienna LTE Simulator es uno de los ejemplos más significativos*	**Simulación de red** *ns-3, junto con la extensión LENA es una de las alternativas más relevantes*
CARACTERÍSTICAS ESCENARIO	*Complejidad del escenario: # de usuarios y estaciones base*	◔ Debido al gran nivel de detalle de estas herramientas, el número de elementos el bastante bajo, normalmente un elemento de acceso y un conjunto de usuarios[11]	◕ Se suelen asumir alsgunas simplificaciones, de modo que el número de elementos suele ser mayor	◑ El tiempo de simulación requerido para analizar escenarios grandes es normalmente inaceptable [12], posibles alternativas usando paralelización [13]
	Dimensión temporal: tiempo que puede ser simulado y posibilidad de estudiar la evolución de servicios	◔ Debido a la carga computacional [14], el tiempo simulado es bastante reducido, sin necesidad de mantener evolución de servicios	◔ El uso de entornos de desarrollo pesado (Matlab) normalmente impide tiempos largos de simulación	◕ Normalmente se considera la evolución de servicios, sin embargo, el tiempo de cómputo para simulaciones largas es muy elevado
	Precisión: grado de precisión de los modelos usados	● El modelado detallado de las capas inferiores es su principal objetivo,por lo que la precisión es muy alta	◕ Se asumen algunas simplificaciones aunque implementaciones disponibles siguen las especificaciones del 3GPP	◕ Aunque los modelos pueden ser simplificados, la implementación de los protocolos es bastante precisa
MARCO TECNOLÓGICO	*Cambio de arquitectura: posibilidad de añadir y soportar nuevos paradigmas de red: SDN and NFV*	◔ Como solución a nidel de enlace, no se consideran problemáticas de arquitectura	◕ Algunas de las posibilidades de las nuevas funcionalidades de red (tighter cooperation schemes) normalmente se pueden modelar	● Aunque la implementación puede ser costosa, la integración de nuevas opciones de arquitectura son normalmente posibles
	Soporte de diferentes tecnologías y soluciones/técnicas	◑ Se encuentran bastante limitadas a las funcionalidades iniciales. La integración de diferentes tecnologías es normalmente compleja	◕ Normalmente tienen flexibilidad para incorporar nuevas técnicas debido a las simplificaciones de capas inferiores	◕ Las plataformas de simulación de redes son bastante flexibles, y permiten la integración de diferentes tecnologías y técnicas nuevas
	Modelado de servicios. Si se asume condiciones de saturación o carga constante	◔ No se presta mucha atención al modelado de servicios, normalmente se centra en cómo los paquetes llegan a su destino en las capas inferiores	◑ Presentan caracterización básica de servicios, aunque normalmente se asume carga constante o fullbuffer [15]	● Modelado de servicios relativamente avanzado. Permite incluso el uso de aplicaciones y servicios reales
OTROS ASPECTOS	*Propósito específico Vs.genérico y curva de aprendizaje*	◕ Como su ámbito de aplicación está bastante delimitado, la curva de aprendizaje es relativamente corta	◑ Aunque más específicos que los simuladores de red, no todos sus componentes son siempre de interés	◔ Normalmente son grandes entornos de propósito general, por lo que requiere bastante tiempo de aprendizaje antes de poder realizar análisis
	Uso de metodologías complementarias. Técnicas de optimización	◔ Normalmente se centran en analizar el rendimiento de una técnica concreta, y típicamente no buscan el rendimiento óptimo	◑ Aunque no está entre sus objetivos principales, las técnicas de optimización se pueden integrar	◑ La arquitectura del simulador ofrece una visión de conjunto, lo que permitiría aplicar estrategias de optimización global

En base a las características de las metodologías de simulación que se han presentado hasta el momento, surge como cuestión a tener muy en cuenta, qué herramienta se debería usar en el caso de este Trabajo Fin de Máster. Para ello, hay que tener en cuenta el tipo de escenario y topología de red de interés, ya que no existe una solución que sea la idónea para todos los casos.

Como consecuencia, en muchas ocasiones se opta por desarrollos propietarios, lo que requiere invertir un tiempo muy elevado en el desarrollo de los mismos. Además, dado que se trata de soluciones ad-hoc, es complicado replicar los experimentos, así como integrarlos en otros entornos.

3.2. Metodología de simulación: GWNSyM

Con el objetivo de buscar la solución más adecuada para el tipo de análisis que se acometen en este trabajo, se ha decidido implementar un desarrollo propietario, GWNSyM. Esta plataforma proporciona una solución flexible para la simulación de sistemas complejos. Además, se ha diseñado de modo genérico, para que sean fácilmente aplicables nuevas funcionalidades o soluciones de red, ya que su estructura es completamente modular.

Para entender la utilización de esta herramienta en este trabajo, es importante presentar brevemente la situación de las redes de comunicación actuales, que hacen que concluyamos en la necesidad de un entorno que permita analizar escenarios donde se utilice virtualización de funciones de red en arquitecturas C-RAN.

Como ya se ha comentado en la Introducción de este tabajo, las previsiones que se encuentran en [16], preveen que la demanda de tráfico en redes móviles inalámbricas se incrementará de forma notable en los próximos años. Esto es debido a la consolidación de servicios de gran capacidad, tales como el streaming de vídeo o los juegos online, que compartirán los recursos de las redes con otros más tradicionales, como la navegación web o las descargas de ficheros.

Bien es cierto que a día de hoy, la consolidación de las tecnologías 4G no se ha producido totalmente, pero la comunidad investigadora tiene que ir un paso por delante, y ya está dirigiendo sus esfuerzos a la definición de las bases del 5G, que daría soporte a la previsible heterogeneidad de servicios, como se ha explicado en el Capítulo de introducción teórica del C-RAN.

Toda previsión de futuro, pasa por la coexistencia de redes de diferentes tecnologías, y que la cooperación entre ellas se lleve a cabo de manera natural. Por ejemplo, se espera que las estrategias de densificación mediante small-cells jueguen un papel muy importante en los próximos años [17], ya que pueden proporcionar un notable aumento de la capacidad. Otras técnicas que se han incluido recientemente en las especificaciones del 3GPP son las referidas a la cooperación entre elementos de red, o técnicas CoMP, o el desacoplamiento de los enlaces ascendente y descendente de las conexiones, o DUDe.

Además de las soluciones que se sitúan en las capas inferiores de las redes celulares, centradas en la gestión de recursos, en las capas superiores las técnicas de virtualización de red [18], NFV y SDN se presentan como elementos clave de los despliegues de red en los próximos años [19]. A pesar de los claros avances que proporcionan estas soluciones, también originan nuevas problemáticas, que requieren un estudio y análisis apropiados como los que se realizarán en este trabajo. Para ello, la comunidad investigadora se centra habitualmente en escenarios específicos y casos de uso concretos.

Para poder analizar estos nuevos conceptos de red, es necesaria la utilización de una herramienta que permita la creación de escenarios concretos, para su posterior análisis. Es por esto, que a continuación se presenta el funcionamiento de la herramienta GWNSyM, que permite incluir todas las funcionalidades que se han comentado con anterioridad.

3.2.1. Funcionamiento

Como principal objetivo de GWNSyM, se busca proporcionar una serie de abstracciones que faciliten el modelado de redes para su posterior simulación. Por otro lado, en lo que se refiere a la metodología utilizada para realizar el análisis, no se utiliza una simulación basada en eventos como es común, sino que se ha decidido utilizar un procedimiento basado en fotografías del sistema. De esta manera, cada fotografía que se realice en el análisis, representa un instante discreto en el tiempo. En dicho instante, se aplican todos los modelos implementados sobre los elementos de red pertinentes y en un orden establecido, según el escenario que se desea ejecutar. Además, el estado resultante de una fotografía, por ejemplo información sobre qué servicios se han podido cursar correctamente, es usado para alimentar la siguiente fotografía. Así, la herramienta proporciona la posibilidad de capturar la memoria del sistema, lo que es especialmente importante para analizar la evolución de los servicios. En el caso concreto de este trabajo, esa posibilidad que ofrece la herramienta, no va a ser explotada en gran medida, ya que no se usarán servicios durante el análisis.

El entorno de simulación GWNSyM ha sido implementado como un conjunto de librerías en C++, teniendo como uno de sus objetivos principales la reutilización de código, que puede darse en dos sentidos:

- Código generado en el entorno GWNSyM se pueda reusar en otros entornos: se decidió no imponer restricciones de herencias a las clases C++ que implementan las diferentes entidades y modelos del simulador, incorporando un interfaz mínimo que asegura la interacción de elementos GWNSyM. Este interfaz garantiza la compatibilidad de los diferentes elementos implementados dentro de GWNSyM, fomentando, de este modo, la separación de los modelos del sistema en el que se ejecutan.
- Código existente pudiera integrarse dentro del simulador: se ha añadido una funcionalidad de wrapper, que permite la integración de código existente dentro del entorno de simulación, incluyendo el interfaz requerido.

A continuación, se detallarán con más profundidad tanto los diferentes elementos que se definen en una simulación con GWNSyM, como la especificación de un escenario con la herramienta.

Elementos de simulación

Se han definido dos elementos esenciales que constituyen los escenarios de simulación en la herramienta GWNSyM, cuya principal característica es que sean dinámicos y no limiten el comportamiento de la herramienta a ninguna tecnología ni sistema en particular:

- *Tipos* : definen la estructura de un elemento de red de forma general, junto con una configuración concreta. En este sentido, un elemento puede abarcar desde dispositivos de usuario a operadores, pasando por servicios o elementos virtuales. Además, de acuerdo a su configuración, un *Tipo* puede agregar elementos de otro *Tipo*, de forma que se pueda definir la composición de cada elemento de red como una combinación de *Tipos*. La instanciación de elementos de un determinado *Tipo* define el conjunto de elementos correspondiente. A modo de ejemplo, la Figura 3.1 ilustra la creación de *Tipos* en GWNSyM para un caso genérico. Como se puede observar, el sustrato de los *Tipos* consiste en clases C++ que, junto a una configuración concreta, da lugar a un *Tipo*. Así, la Figura 3.1 muestra cómo una misma clase C++ C1 da lugar a dos *Tipos* (T1 y T2), en función de la configuración que se le aplica. Además, de acuerdo a la configuracion, las instancias de cada *Tipo* pueden dar lugar a diferentes agregaciones.

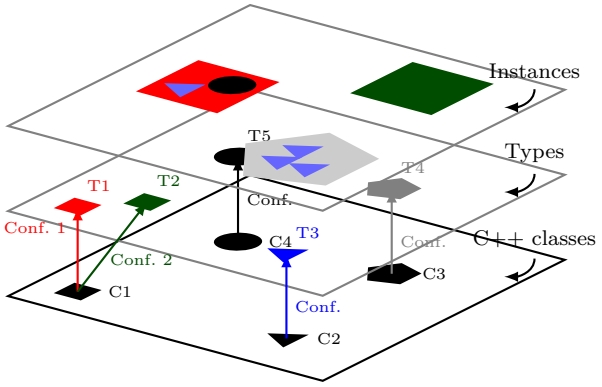


Figura 3.1: Modelo de instanciación de GWNSyM

- *Acciones*: representan un modelo particular que se va a aplicar a uno o más conjuntos de *Tipos*. En general, las acciones representan comportamientos del sistema en sentido amplio, pudiendo abarcar desde fenómenos físicos, tales como modelos de propagación, a políticas concretas, como selección de acceso. Cada acción toma como parámetros uno o más conjuntos de *Tipos*, y se ejecutan de forma secuencial en cada fotografía del escenario. De este modo, las acciones tienen lugar en el bucle más interno de la simulación.

Por otro lado, se tienen ciertas excepciones, en las que una acción determinada únicamente tiene sentido al inicio o al final del análisis, como podría ser el despliegue de elementos de acceso estáticos. Estos supuestos se han tenido en cuenta definiendo dos categorías de acciones, *Pre-Acción* y *Post-Acción*, que se ejecutan al principio y final del experimento, respectivamente.

En este sentido, una instancia del experimento, representa el análisis de un escenario concreto, sujeto a una configuración específica. Cada experimento contiene dos bucles: uno exterior y otro interior. El primero realiza el paso de fotografía a fotografía, actualizando el estado de la red de acuerdo al resultado de la fotografía anterior. Por su parte, el segundo bucle se encarga de aplicar los modelos, o acciones, correspondientes a los elementos de red, o tipos definidos, dentro de una fotografía. En la Figura 3.2 se puede observar el comportamiento descrito con anterioridad.

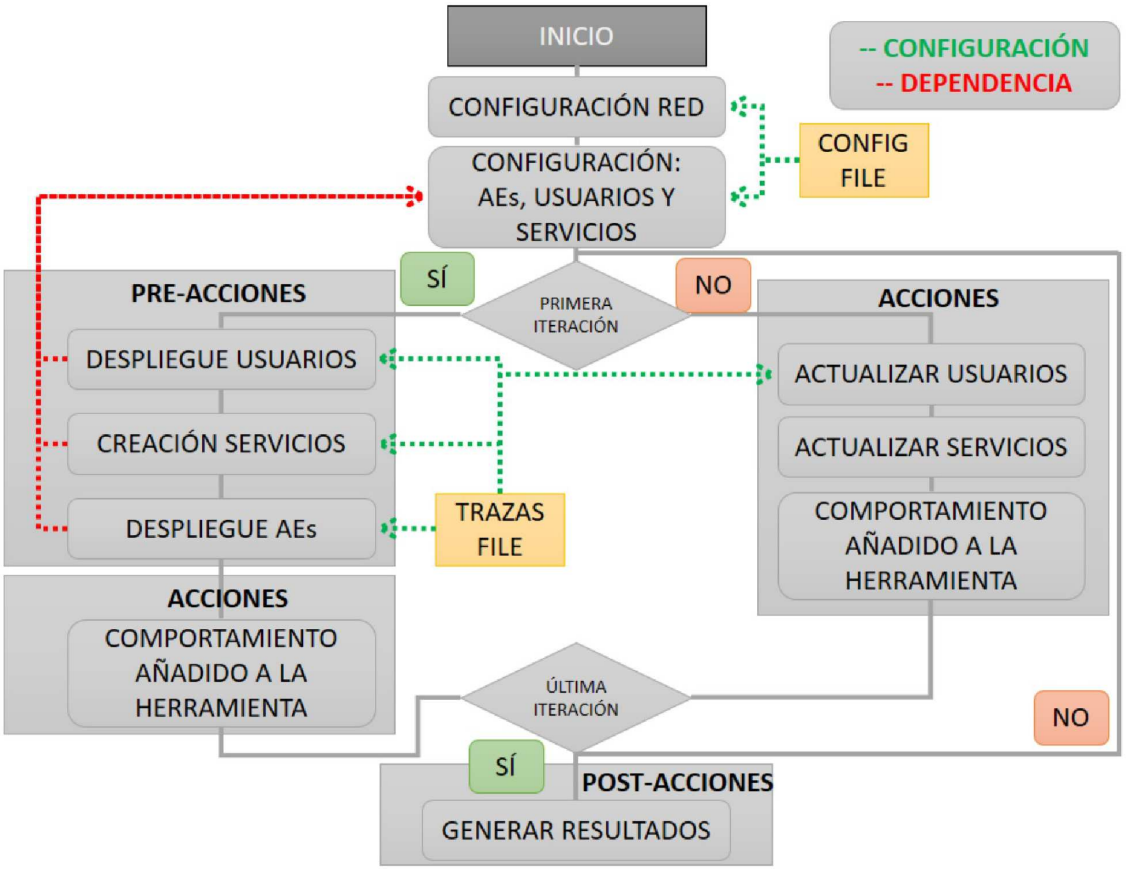


Figura 3.2: Funcionamiento de la herramienta GWNSyM

Como complemento para la ilustración de la metodología general, se presenta el Algoritmo 1. A continuación, quedan descritas una serie de indicaciones para la interpretación del mismo:

- Se definen los *Tipos* y se crean sus instancias, dando lugar a los conjuntos previamente mencionados, de acuerdo a su configuración
- Una vez que todos los elementos del sistema han sido instanciados, se ejecutan las acciones definidas como *Pre-Acciones*, como podrían ser el despliegue de los elementos de acceso del sistema
- A continuación, se ejecutan las *Acciones*, que se encuentran en un bucle más interno, y se ejecutan para todas las fotografías que se analicen del escenario, según el orden que se indique en el mismo. En el caso de este trabajo, consistiría en ejecutar los algoritmos implementados
- Por último, se aplican las *Post-Acciones*, normalmente encargadas de extraer resultados y generar trazas

Algoritmo 1 Flujo de simulación genérico

```
1: Definición de Tipos
2: Configuración
3: Instanciación y agregación
4: T ← Sets
5: A_pre ← Pre-Actions
6: A ← Actions
7: A_post ← Post-Actions
8: i = 0
9: n ← # Snapshots
10: for b ∈ A_pre do
11:     Ejecutar Pre-Action b(M_b ⊆ T)
12: end for
13: while i < n do
14:     for a ∈ A do
15:         Ejecutar Action s(M_s ⊆ T)
16:     end for
17: end while
18: for e ∈ A_post do
19:     Ejecutar Post-Action e(M_e ⊆ T)
20: end for
```

Ejemplo de definición de un escenario

A modo de ejemplo, en los siguientes fragmentos de Código 3.1, 3.2, y 3.3 se ilustra los principales aspectos de la definición de un escenario en la herramienta.

Como se puede observar en el Código 3.1, el primer paso consiste en el registro de *tipos* dentro del sistema. Todos los *tipos* consisten en dos objetos C++: el elemento de red y su configuración. Por ejemplo, el *tipo* `USER` se define en base a los objetos `User` y

`UserConf`. Del mismo modo se registrarían los tipos para el resto de elementos de red que están definidos.

Antes de instanciar los elementos de cada *tipo*, el sistema comprueba aquellos que debe agregar, ya que unos tipos pueden estar incluidos en otros, a través del nombre del *tipo* correspondiente. Como se muestra en el Código 3.2, la configuración del *tipo* `USER` indica que agrega una instancia del *tipo* LTE_UE, de modo que se instancia un elemento del *tipo* LTE_UE por cada elemento del *tipo* `USER`. Por otro lado, también sería un ejemplo muy visual de las agregaciones de elementos, en este caso de estaciones base $ENBS$, que contiene los elementos de acceso de tipo Pico, Micro y Macro de todos los clusters del escenario.

Una vez se tienen establecidos los *tipos* y sus agregaciones, se registran las *acciones* que definen el comportamiento del sistema, tal y como se muestra en el Código 3.3. Es importante destacar que se ha dotado al entorno de simulación de un módulo que permite la búsqueda de los elementos instanciados, de forma que el paso de conjuntos de elementos a las *acciones* se lleva a cabo de una manera muy flexible. Por ejemplo, los elementos indicados en el código como `MACRO::*::CELL` se corresponden con un elemento de *tipo CELL* agregado en todos los de *tipo MACRO*.

```
gnsm::System system;
...
system.AddType<User, UserConf>("USER");
system.AddType<LteUe, LteUeConf>("LTE_UE", {Params});
system.AddType<LteCell, LteCellConf>("CELL", {Params});
system.AddType<LteEnb, LteEnbConf>("MACRO", {Params});
...
```

Código 3.1: Definición de *Tipos*

```
UserConf::ReadInnerConf(void) const
{
    return{{"LTE_UE", 1}}; // read from configuration
}
```

Código 3.2: Agregación

```
system.PreAction<MacroDeploymentr>({"MACRO"}, {Params.});
...
system.Action<LteScan>({"USER", "MACRO"}, {Params.});
...
system.PostAction<MacroLoad>({"MACRO::*::CELL"}, {Params.});
...
system.Run();
```

Código 3.3: Acciones

Todo lo anterior sirve para entender como funciona el núcleo de la herramienta que se ha creado para la simulación de este tipo de redes. En el caso de este trabajo de Fin de Máster, no se van a utilizar todas las posibilidades que la herramienta ofrece, debido a que el estudio se va a centrar en el fronthaul del escenario. Por todo esto, no se va a prestar atención a todos los aspectos que tengan que ver con los usuarios de la red, ni

a los servicios que vayan a cursar, porque el análisis no se va a centrar en la parte de propagación radio, sino en la interconexión de estaciones base.

Por lo cual, los *Tipos* de elementos que se van a utilizar en el ámbito de este trabajo se limitan a:

- Estaciones base
- BBUs

Y respecto a las *Acciones* que se van a aplicar a estos *Tipos* se diferencian :

- Pre-Acciones : despliegue de las estaciones base por primera vez en el escenario
- Acciones : resolución de los algoritmos de optimización y minimización de coste que se explican en los siguientes Capítulos
- Post-Acciones : extracción de resultados e interpretación de los mismos, también descritos en los siguientes Capítulos

Capítulo 4

Localización óptima de BBUs

En esta primera parte, el objetivo planteado es la creación de un algoritmo que resuelva el problema de la P-Mediana. En este caso concreto, se pretende minimizar el coste que supondría introducir p BBUs en un escenario donde se tengan desplegadas estaciones base de diferentes tipos, para realizar un estudio del C-RAN para el futuro 5G.

A continuación, se presentan los conceptos teóricos necesarios para la comprensión del problema de la P-Mediana, así como el algoritmo greedy que permite su resolución.

Por otro lado, se ha creado un escenario de prueba para este algoritmo, en sus dos posibles configuraciones de funcionamiento:

- Cuando se basa en una función de coste que minimiza la distancia total de los enlaces desplegados en el escenario.
- Cuando la función de coste se basa en los costes de las diferentes tecnologías que se pueden utilizar en el mercado hoy en día.

4.1. P-Mediana

El problema de la P-Mediana [20], consiste en localizar 'P facilities' para un conjunto de usuarios, donde se minimiza la suma de las distancias ponderadas. Todo esto, llevado al problema que se plantea en este trabajo, consiste en situar 'P BBUs' en un escenario, y conectarlas con el conjunto de las estaciones base desplegadas en el mismo, de modo que la longitud/coste de los enlaces sea mínima.

Como casi todos los problemas de localización, está clasificado como un problema NP-hard. Debido a esto, si se considera el número total de posibles soluciones, se tiene:

$$\binom{N}{P} = \frac{N!}{P!(N-P)!}$$

Por lo que, para resolverlo eficientemente se utilizan diferentes métodos heurísticos, que no realizan una búsqueda exhaustiva de todas las posibles soluciones, disminuyendo en gran medida el tiempo de cómputo necesario para resolver el problema.

En el problema que se va a resolver, de naturaleza combinatoria y de optimización global de la función de coste, se tiene como objetivo llegar a:

$$min\{f(\chi)|\chi \in X\}$$

donde f(x) es la función que se pretende minimizar, y X el conjunto de posibles soluciones. Una solución $\chi*$ es óptima si cumple :

$$f(\chi*) < f(\chi) \quad \forall \chi* \in X$$

La P-mediana es un problema combinatorio y se puede plantear definiendo dos variables de decisión:

- $y_j = 1$: si la 'BBU' se encuentra en la ubicación de j, e igual a 0 en otro caso
- $\chi_{i,j} = 1$: si la estación base i está siendo servida por la 'BBU' j, e igual a 0 en otro caso

El problema de la P-Mediana con una solución, sigue la siguiente formulación y como se puede ver, está sujeto a una serie de restricciones:

$$min \sum_i \sum_j d_{i,j} \cdot \chi_{i,j}$$

Sujeto a :

$$\sum_i \chi_{i,j} = 1; \quad \forall i$$
$$\chi_{i,j} \leq y_j \quad \forall i,j$$
$$\sum_j y_j = p$$
$$\chi_{i,j}, y_j \in \{0,1\}$$

4.1.1. Algoritmos de resolución de este tipo de problemas

Si se utilizase un algoritmo exacto que solucione el problema, en caso de que éste tenga solución, se asegura encontrar la solución óptima. Sin embargo, por su ineficiencia computacional, hay que hacer una búsqueda exhaustiva, no se recomiendan. Por otro lado, están los algoritmos heurísticos, que encontrarán mucho más rápido una solución que estará muy cerca de ser la óptima.

A lo largo de este trabajo se ha realizado, en primera instancia, una comparativa de una solución basada en algoritmos heurísticos y exactos y, dependiendo de la magnitud del problema que se plantease, el tiempo invertido en encontrar una solución mediante el algoritmo exacto, crece exponencialmente, mientras que el crecimiento del tiempo en resolver el problema con un algoritmo heurístico es lineal.

De este modo, se hace la primera selección para resolver el problema de la P-mediana. Aunque no asegure ser la solución óptima, se elige un algoritmo heurístico para la resolución.

Los clásicos algoritmos heurísticos para resolver el problema de la P-mediana son: Greedy, Stingy, Dual ascent, Alternate, Interchange y Composite heuristics. Los primeros tres son de naturaleza constructiva, mientras los otros dos, necesitan una solución feasible inicialmente.

En el caso de este trabajo, se ha elegido implementar en la herramienta un algoritmo greedy para encontrar la solución a la P-mediana, propuesto por Kuehn y Hamburger (1963). Consiste en empezar con un conjunto de 'facilities' vacías, para luego resolver el problema 1-mediana P veces, eligiendo la localización de las 'facilities' una a una, hasta llegar a las p establecidas por las condiciones iniciales del problema. Se van eligiendo esas 'facilities' secuencialmente, según cuál sea la que más reduce el coste.

4.2. Algoritmo creado para la minimización del coste

Como bien se ha explicado con anterioridad, para hacer uso de este algoritmo se ejecuta P veces una 1-mediana, reduciendo en gran medida el número de opciones a explorar. A continuación se presenta el conjunto de módulos que se han añadido al núcleo de la herramienta para resolver el problema.

En este caso, se sigue una estructura anidada, es decir, no se llama a una clase, que finaliza antes de llamar a otra, sino que sólo se va a llamar a una primera clase, que para resolver el problema, llamará internamente a otra, y así sucesivamente hasta llegar al cálculo básico de la distancia origen-destino.

A la clase inicial, que tiene una filosofía como la que se ve en la Figura 4.1, se le pasan como parámetros de entrada:

- Número de 'facilities' que va a tener el problema, es decir, el número de BBUs que se pretende colocar en la red
- Tipos de estaciones base que tienen la capacidad de ser 'facilities', ya que no todas tienen porqué tener la capacidad de actuar como BBUs
- Se requiere de una variable interna, que indica el número de estaciones base que han sido ya asignadas como BBUs

Se arranca el algoritmo y mientras el número de estaciones asignadas como BBUs sea menor que el número total de BBUs del escenario, se ejecuta la función que cálcula el coste, que resolverá una 1-mediana (Figura 4.2) como se ha comentado con anterioridad.

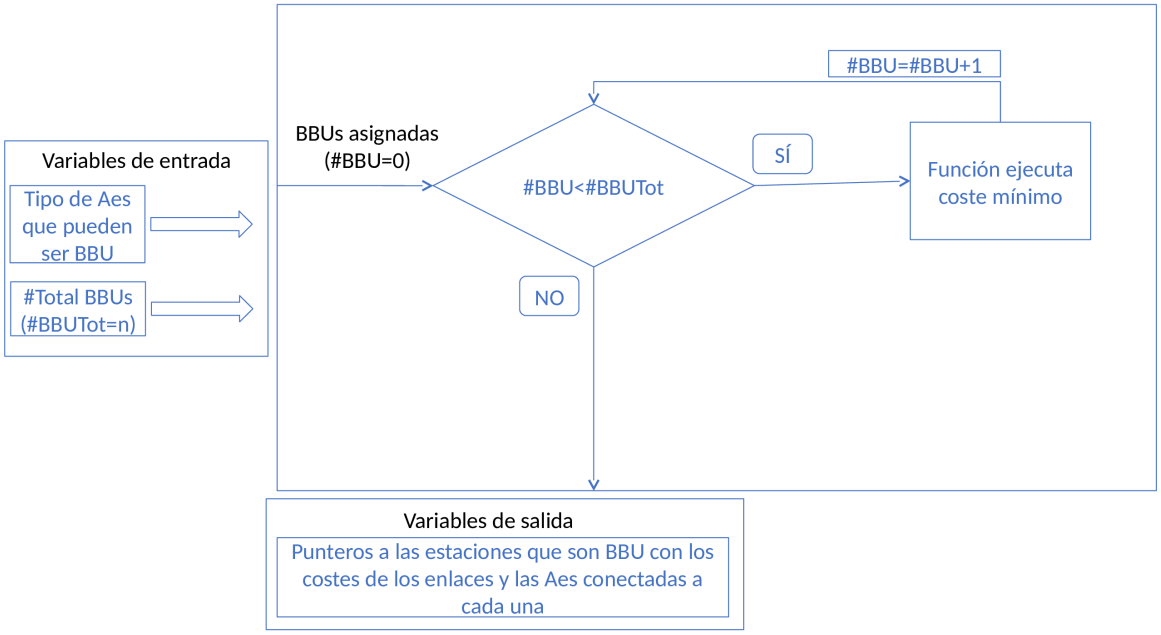


Figura 4.1: Estructura de la clase que resuelve la P-mediana

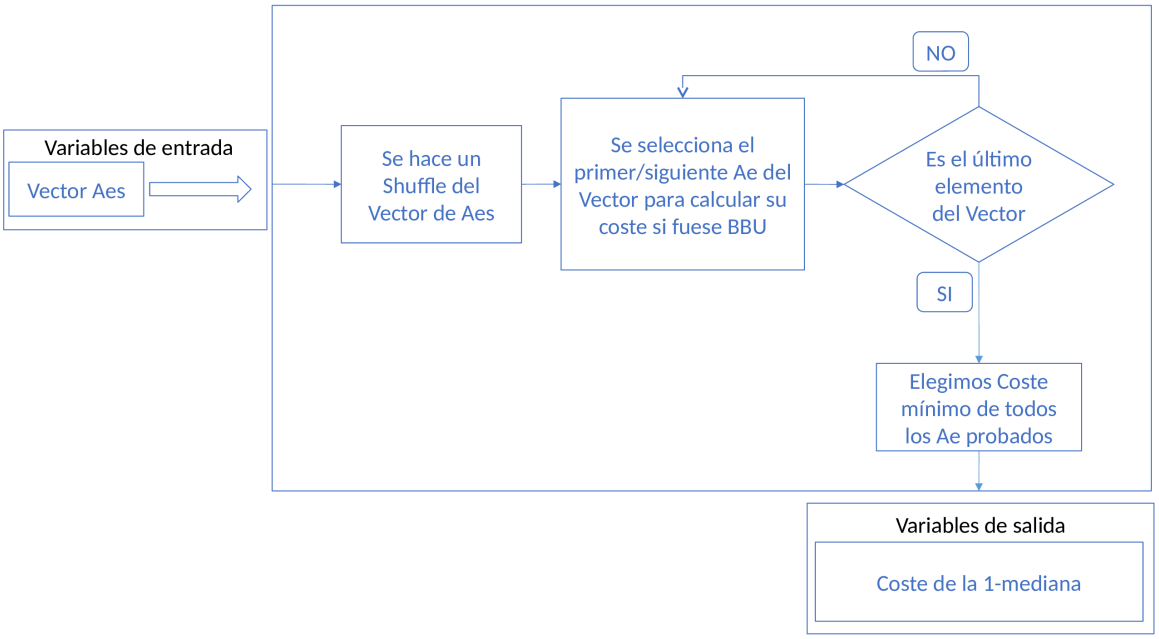


Figura 4.2: Estructura de la implementación de la 1-mediana

En este paso del desarrollo del algoritmo (solución 1-mediana) se necesita simplemente un vector de punteros con la información de las estaciones base. En primer lugar, se realiza un shuffle al vector, de tal manera que se 'barajan' las estaciones base, para

tener un reparto más justo. Esto es así, debido a que cada BBU va a tener un límite de estaciones base a las que va a poder servir, y el orden con el que se procesan las BBUs influye en el coste. Para poder beneficiarnos de la utilidad del shuffle, se realizarán n análisis para las mismas posiciones de las estaciones base. Es decir, se barajeará el vector de estaciones base n veces, y se calculará la p-mediana para cada una de ellas, eligiendo a posteriori la mejor solución que se haya obtenido.

Una vez que se tiene el vector de estaciones base desordenado, se van seleccionando una por una, como posible BBU, y se calcula el coste que supondría, siempre y cuando pertenezcan al tipo de estaciones base que pueden ser BBU (dado por configuración). A continuación, se escoge la de menor coste y se selecciona como BBU.

Para saber cómo se calcula ese coste para cada BBU, se debe de entrar en un nivel más interno del proceso, una nueva función que lo calcula. De este modo, se sigue la estructura que se indica en las Figuras 4.3, 4.4, 4.5, 4.6 y 4.7.

En primer lugar, los parámetros de entrada que se necesitan para calcular el coste que supondría que la i-ésima estación base fuese BBU son:

- Puntero a la i-ésima estación base para poder acceder a todos sus parámetros y tipos de configuración
- Matriz de distancias entre todas las estaciones base
- Punteros a las BBUs ya seleccionadas

En esta etapa, para comprender el funcionamiento del algoritmo se ha puesto un ejemplo, donde ya han sido seleccionadas como BBUs las estaciones base 1 y 4, y se está probando el coste que supondría que la tercera estación base seleccionada como BBU fuese la 3. Para realizar esta fase han de calcularse las distancias para todas las estaciones base a las estaciones 1, 3 y 4.

Se va escogiendo, de esas tres opciones, la mínima distancia en cada caso, siempre y cuando no se supere el número máximo de estaciones conectadas a la BBU, en cuyo caso, se escogería la siguiente mejor. Se va realizando el cálculo sucesivo hasta que se han procesado todas las estaciones, momento en el que se devuelven los parámetros de salida (coste para la Ae i-ésima que se está probando), como se ve en la Figura 4.7,a la función que realizó la llamada de ésta.

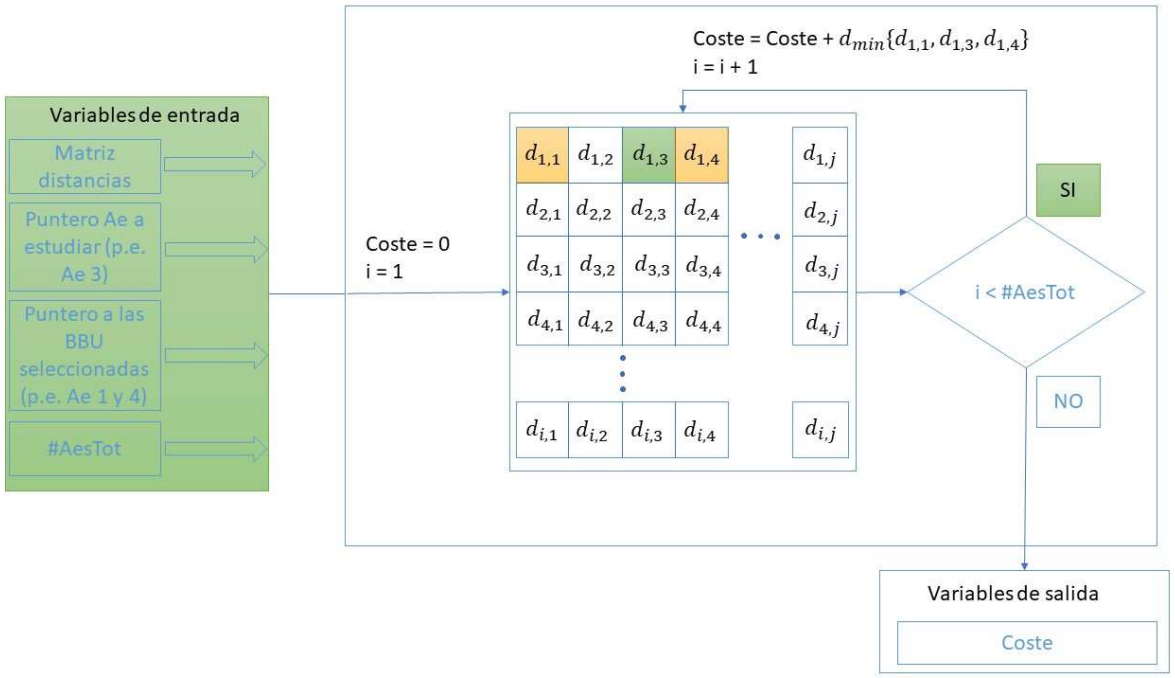


Figura 4.3: Parámetros de entrada y primer paso del cálculo del coste

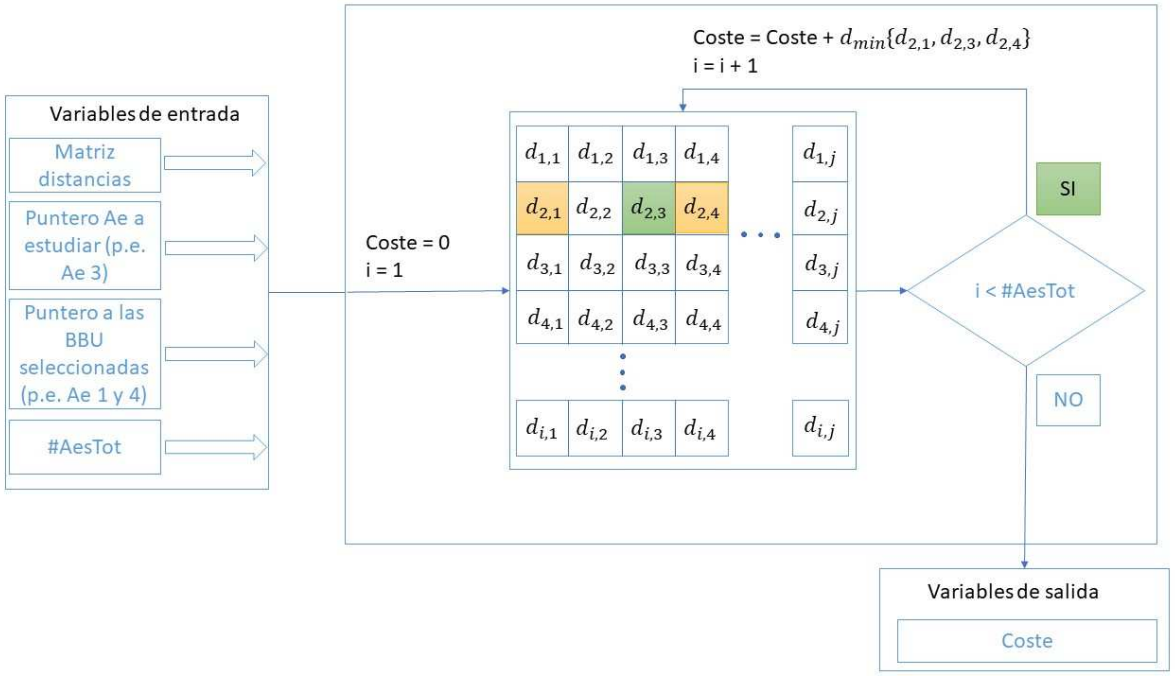


Figura 4.4: Segundo paso del cálculo del coste

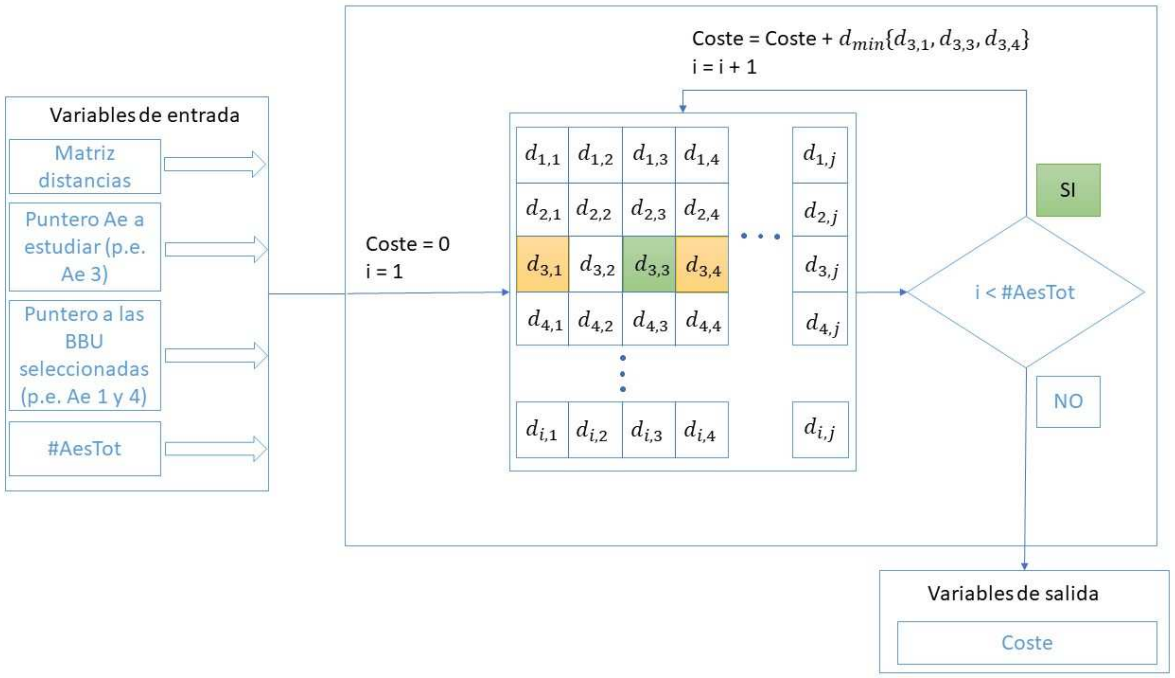


Figura 4.5: Tercer paso del cálculo del coste

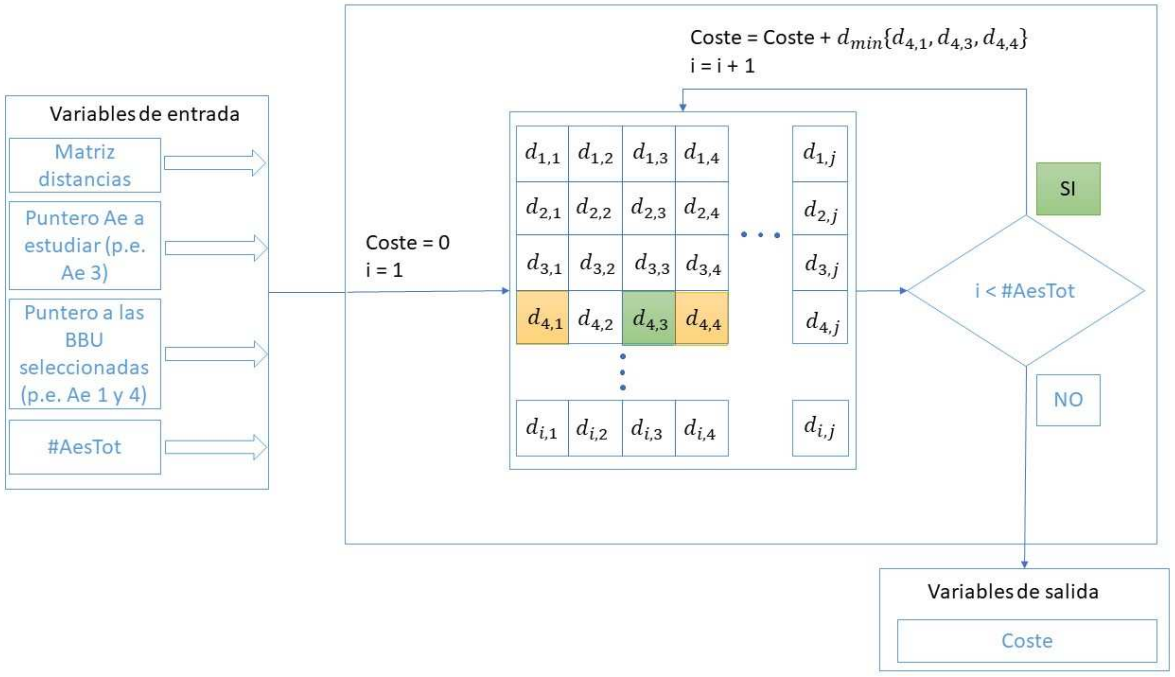


Figura 4.6: Cuarto paso del cálculo del coste

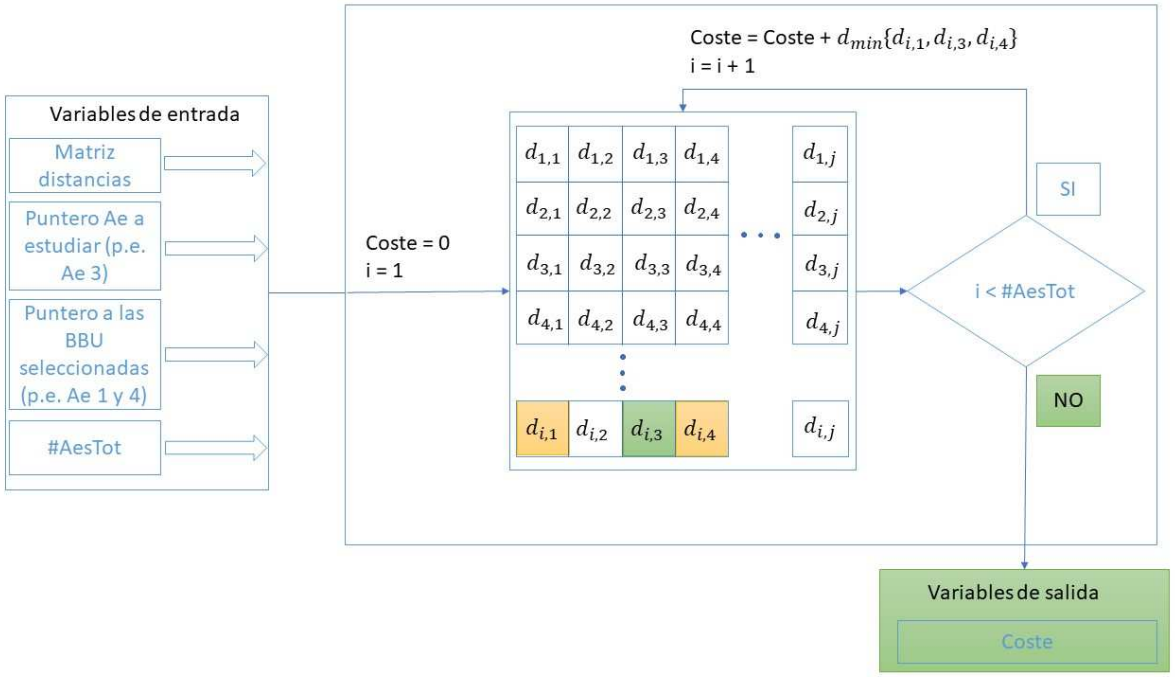


Figura 4.7: I-ésimo paso del cálculo del coste y devolución de parámetros de salida

Se realiza esto, para todas las estaciones base, hasta encontrar la de menor coste, que se selecciona como BBU, y se vuelve a comenzar hasta alcanzar el número de BBUs indicadas por configuración de la red. Todo esto se realiza n veces para cada despliegue de red. Además, para que los resultados sean estadísticamente válidos, se realiza para k despliegues de red diferentes.

Las opciones de configuración que se van a utilizar son:

- Realizar 10 veces el shuffle
- Realizar el estudio en 100 despliegues distintos de estaciones base

4.3. Análisis de resultados

Se plantea un escenario con las siguientes características:

- Dimensiones : 4500m x 4500m
- Estaciones base :
  - Macro estaciones base : 19 en total
  - Micro estaciones base : 95 en total
- Configuraciones : se consideran cuatro configuraciones diferentes según el tipo de despliegue de las estaciones base

- Posibilidad de situar una BBU sólo en las estaciones base de tipo Macro
  - El despliegue de las estaciones base de tipo Macro es aleatorio, cumpliendo con una distancia mínima entre ellas mientras que el despliegue de las micro es completamente aleatorio
  - El despliegue de las estaciones base de tipo Macro se configura en tres anillos, y el despliegue de las micro es completamente aleatorio
- Posibilidad de situar una BBU tanto en las macro como en las micro BS
  - El despliegue de las estaciones base de tipo Macro es aleatorio cumpliendo, con una distancia mínima entre ellas, y el despliegue de las micro es completamente aleatorio
  - El despliegue de las estaciones base de tipo Macro, se configura en tres anillos, y el despliegue de las micro es completamente aleatorio

- Las BBUs se van a situar en las estaciones base que minimicen la solución basada en la distancia/coste mínimo

A continuación, se presentan los resultados que muestran la evolución de la distancia media de enlaces a medida que se va aumentando el número de BBUs. De la Figura 4.8 se concluye que cuando sube el número de BBUs, la distancia de enlace disminuye para las cuatro configuraciones descritas con anterioridad. Bien es verdad que esa comparativa se podría realizar dos a dos, es decir, comparando las configuraciones donde sólo las Macros pueden ser BBUs entre ellas y donde tanto las Macros como las Micros pueden ser BBUs. En este caso, se percibe una diferencia mínima entre las configuraciones, por lo que el tipo de despliegue de las macro estaciones base, apenas afecta al comportamiento.

El objetivo de presentar estos resultados es doble. Por un lado, poder concluir que el algoritmo funciona correctamente, es decir, que la distancia media entre estación-BBU disminuye a medida que el escenario está más poblado de BBUs, lo que se comprueba satisfactorio. Y, por otra parte, determinar cuál es la configuración que incurre en un menor coste para los posteriores análisis.

La configuración que se ha seleccionado es: Macros y Micros pueden tener en su localización una BBU y configuración regular para las BS macro. La diferencia que se encuentra entre la el despliegue regular y el aleatorio no es muy reseñable, debido a que las estaciones base aleatorias, respetan una distancia mínima, para que tengan un cierto sentido al trasladarlo a una red real (Figura 4.9).

Es importante comentar, que esa configuración es la que proporciona una solución con menor coste en distancia, pero también es la que necesita un mayor tiempo de cómputo, debido a que el número de opciones que se tiene que explorar crece notablemente respecto a cualquiera de las soluciones donde se pueden situar una BBU en la posición de las Macro. Aún así, tiene más sentido limitarse a la opción menos eficiente respecto a tiempo de ejecución, pero con una distancia media de enlace menor, ya que el objetivo principal es minimizar el coste.

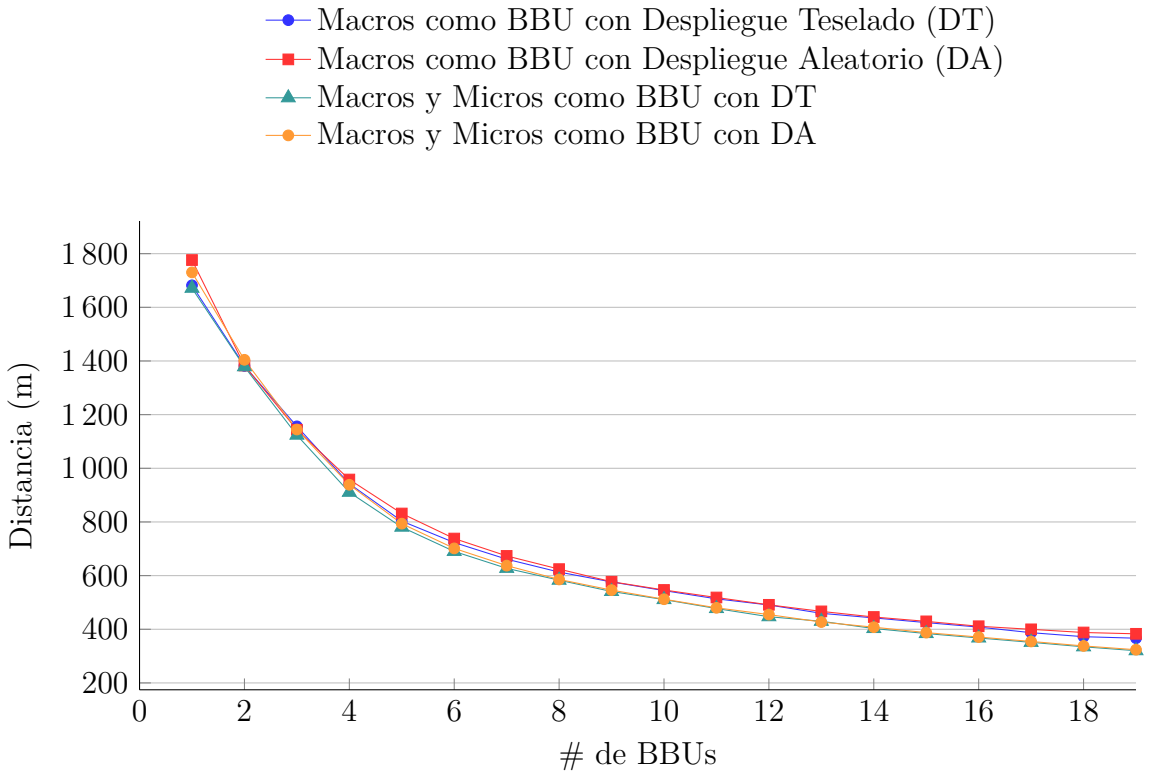
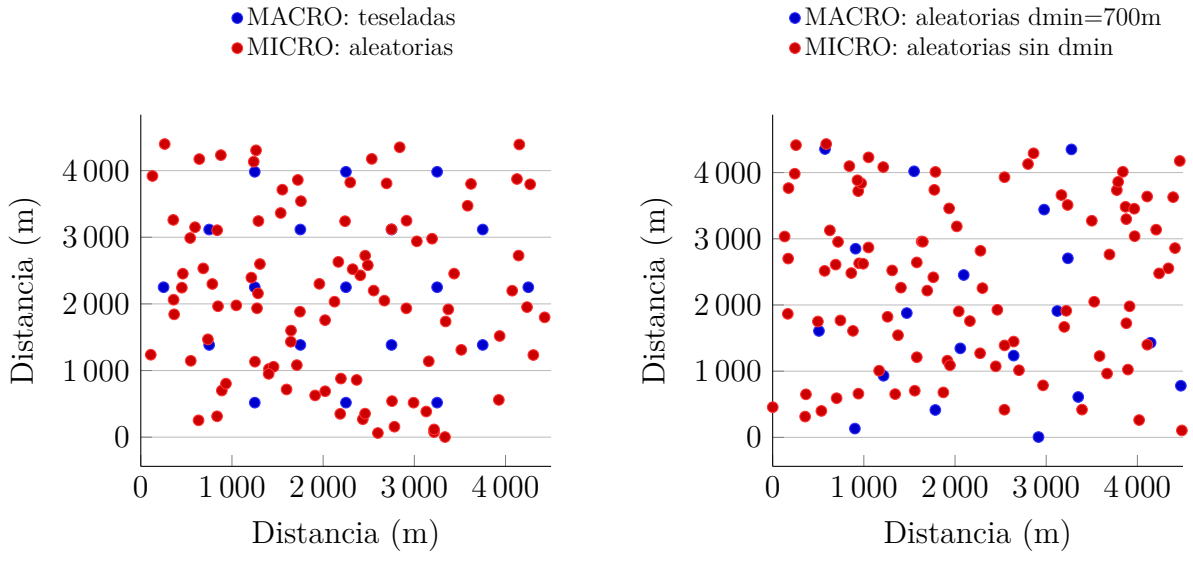


Figura 4.8: Distancia mínima total que calcula el algoritmo según el número de BBUs para las distintas configuraciones



(a) Despliegue teselado

(b) Despliegue aleatorio

Figura 4.9: Despliegue de las estaciones base en el escenario

El siguiente paso, una vez seleccionada la configuración que se va a mantener, es utilizar el mismo algoritmo sobre ella, pero planteando ciertas restricciones que han de

cumplirse para que una estación base pueda estar conectada con una BBU.

En este caso, se plantean las Figuras 4.10 y 4.11 que presentan resultados para un análisis con: 2, 5, 10 y 15 BBUs en el escenario, teniendo como restricción adicional que se puedan conectar a cada BBU como máximo: 3, 6, 9 y 12 estaciones.

Respecto a las conexiones, se ve claramente que el porcentaje de estaciones base conectadas aumenta a medida que la restricción impuesta sobre las estaciones por BBU se relaja. Por otro lado, esas conexiones también aumentan a medida que se habilitan más BBUs en el escenario. La tendencia de conexiones es creciente con las dos características mencionadas, hasta alcanzar el 100 % de estaciones base con enlace hacia una BBU.

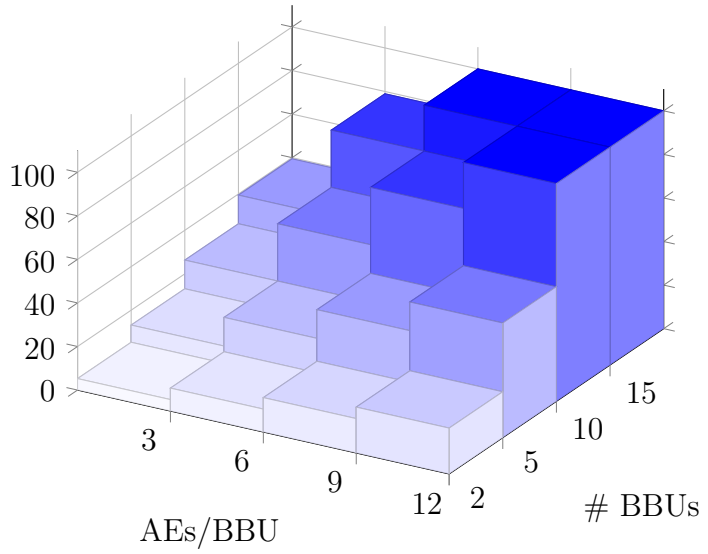


Figura 4.10: Porcentaje de conexiones según el número de BBUs y el número máximo de estaciones base por BBU

En el caso de la distancia media de cada enlace, los resultados son ligeramente diferentes. Salvo en la situación en que el 100 % de estaciones están conectadas, se cumple en todos los casos que, a medida que aumenta el número de BBUs y el número de estaciones por BBU, la distancia de enlace es mayor. Esto es debido a que a medida que las estaciones base que puedan conectarse crecen, estas estarán desperdigadas por el escenario, lo que supondrá un aumento en la distancia media de enlace. En las primeras opciones (cuando hay pocas BBUs y pocas estaciones por BBU) se aprovecha la naturaleza del algoritmo para conectar sólo aquellas que se encuentran muy cerca de las BBUs.

Por otro lado, hay que analizar el comportamiento de las configuraciones donde se alcanza el 100 % de conexiones, debido a que la distancia de enlace cambia la tendencia y disminuye. Esto se debe a que en esos casos, hay opción de conectar a 120, 135 y 180

estaciones. Como en este escenario, el total de estaciones base es de 114, se concluye que la ocupación de los controladores, es diferente según ofrezcan una localización mejor o peor en el escenario. Las BBUs centrales tendrán tendencia a tener una mayor ocupación que las que están en los bordes del escenario.

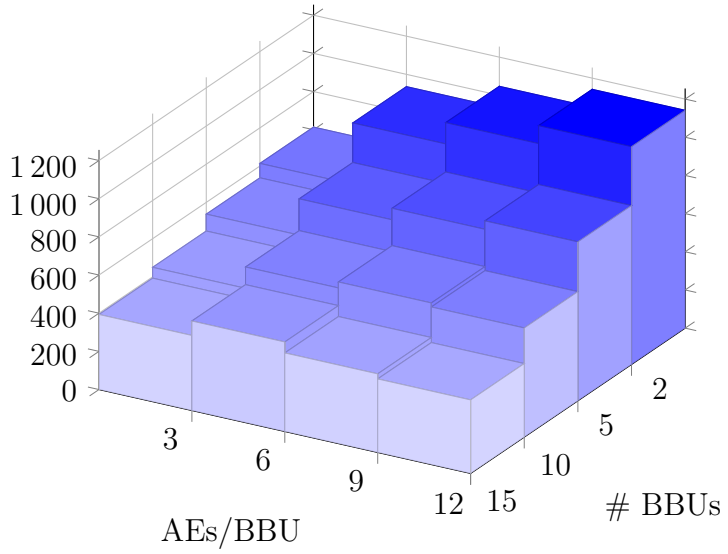


Figura 4.11: Coste medio de enlace según el número de BBUs y el número máximo de estaciones base por BBU

Para justificar esta caída de la distancia de enlace se presenta la Figura 4.13 que, ofrece datos de la ocupación de los controladores según la configuración. Para comprender la interpretación de una caja en una de estas gráficas, se utilizará la Figura 4.12 y la siguiente explicación, para establecer los valores estadísticos que se corresponden con las diferentes líneas de la caja:

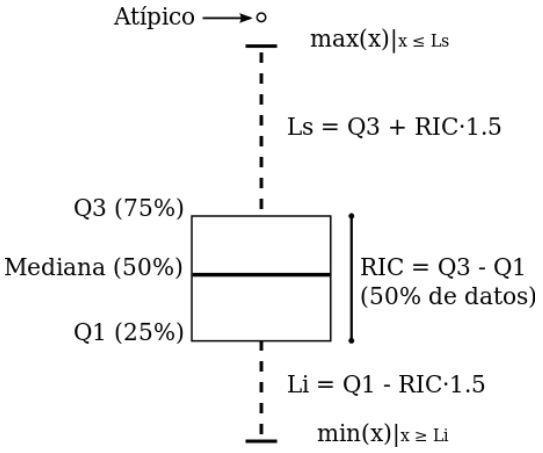


Figura 4.12: Descripción de la interpretación de un boxplot

- La línea superior de la caja se corresponde con el percentil 75 de los datos
- La línea inferior de la caja se corresponde con el percentil 25 de los datos
- La línea intermedia de la caja se corresponde con la mediana, es decir, con el percentil 50 de los datos
- Por otro lado, está el rango intercuartílico, que servirá para saber que valores son atípicos, y se corresponde con la resta de la línea superior de la caja (que se denomina Q3) y con la línea inferior de la caja (que se denomina Q1).
- Las líneas que se extienden por encima y por debajo de la caja, se corresponden con los valores máximo y mínimo de los datos, o con 1.5 veces el valor del rango intercuartílico.
- En caso de que algún valor, se extienda por encima o por debajo de ese valor del Rango intercuartílico (RIC), se denominarán valores atípicos, y se representarán mediante una cruz roja en la representación.

En la Figura 4.13, donde se representa la ocupación de las BBUs cuando hay 15 en el escenario, se observa que, en las dos primeras opciones, no hay variación de ocupación de las BBUs, están al 100 %. En el tercer caso, se observa que un 75 % están ocupadas al 100 %, y en el cuarto caso que un 75 % están ocupadas al 70 %. Por lo que se concluye que, a medida que se permite conectar más estaciones base a cada BBU, el reparto de conexiones es más justo.

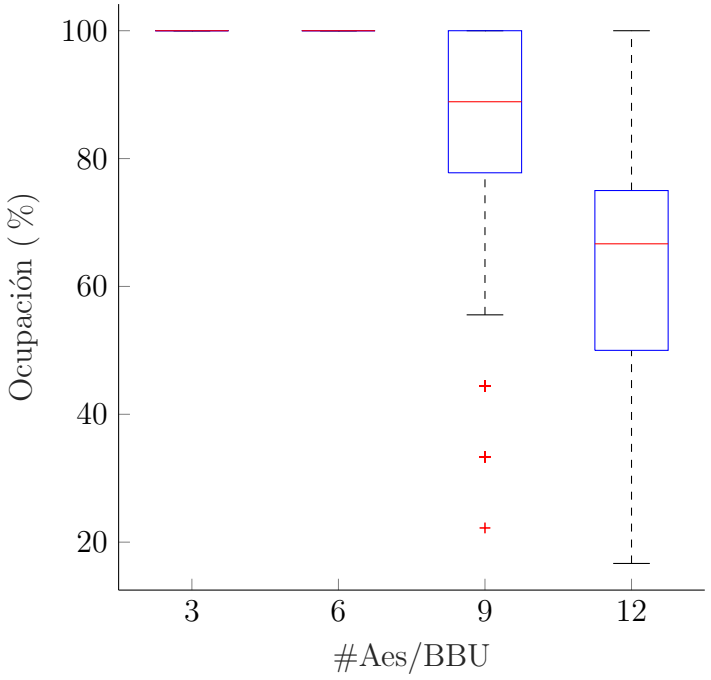


Figura 4.13: Ocupación de las 15 BBUs

El siguiente paso que se plantea, es elegir a priori el split de cada tipo de estación base (en este caso, se ha limitado a los niveles físico y MAC). Por los requisitos de retardo y throughput que tiene cada nivel de split, no todas las tecnologías son apropiadas en todos los enlaces. Concretamente, para el caso de split físico, sólo se puede usar fibra óptica, y en el caso de split en la capa MAC, al ser menos restrictivo, se pueden utilizar tanto enlaces radio como fibra óptica.

El objetivo consiste en introducir al algoritmo que el coste del enlace dependa del tipo de tecnología que se use, teniendo en cuenta las restricciones que impone el nivel de split. Para ello se plantean como opciones:

- Split físico : uso de fibra óptica siempre
- Split MAC : uso de microondas en Macro estaciones base y de ondas milimétricas en el caso de las Micro BS, siempre que convenga ese tipo de enlaces respecto a la fibra óptica

A continuación se presenta la evolución del coste económico que supone un enlace [21] y [22] a medida que aumenta la distancia del mismo (Figura 4.14). Para el caso de las microondas y las ondas milimétricas el coste del enlace (40000€ y 55000€ respectivamente) es fijo independientemente de la distancia. Por otro lado, el despliegue de fibra óptica supone un desembolso fijo de 5000€ para el operador, más una media de 125€ por metro desplegado.

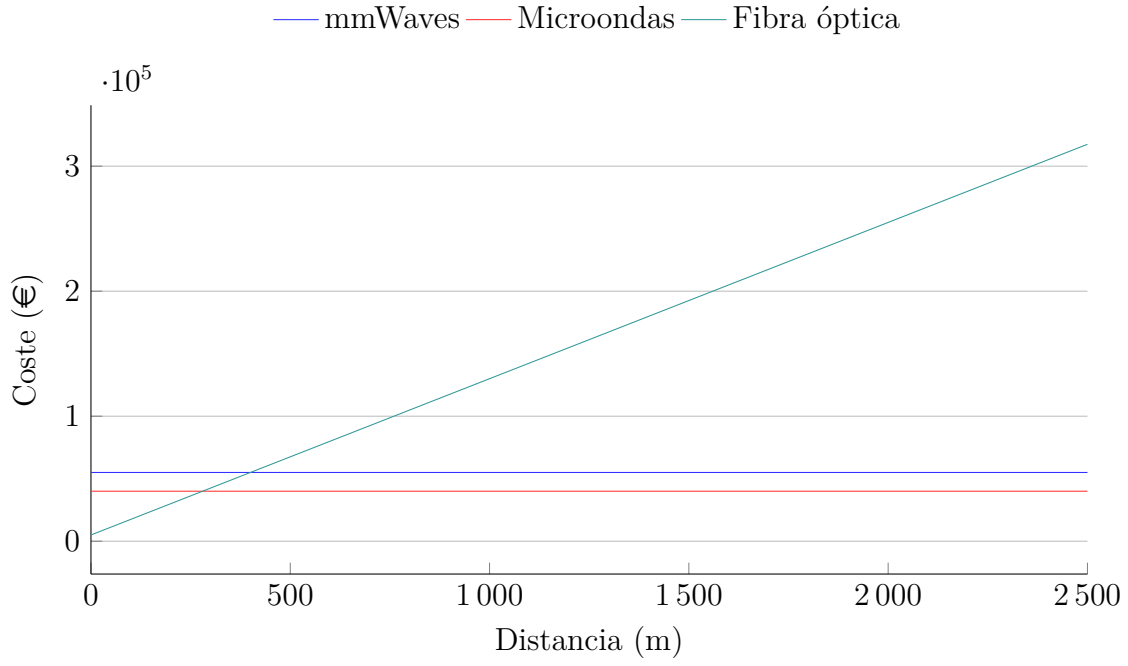


Figura 4.14: Evolución del coste de las tecnologías en respecto a la distancia

Existen artículos como [23], donde se indica que habría la posibilidad de instalar fibra óptica sin la necesidad de abrir una zanja, y se reducirían en gran medida los costes, ya que el mayor peso de la inversión inicial se debe a la obra civil. No se ha contemplado esa opción en este trabajo.

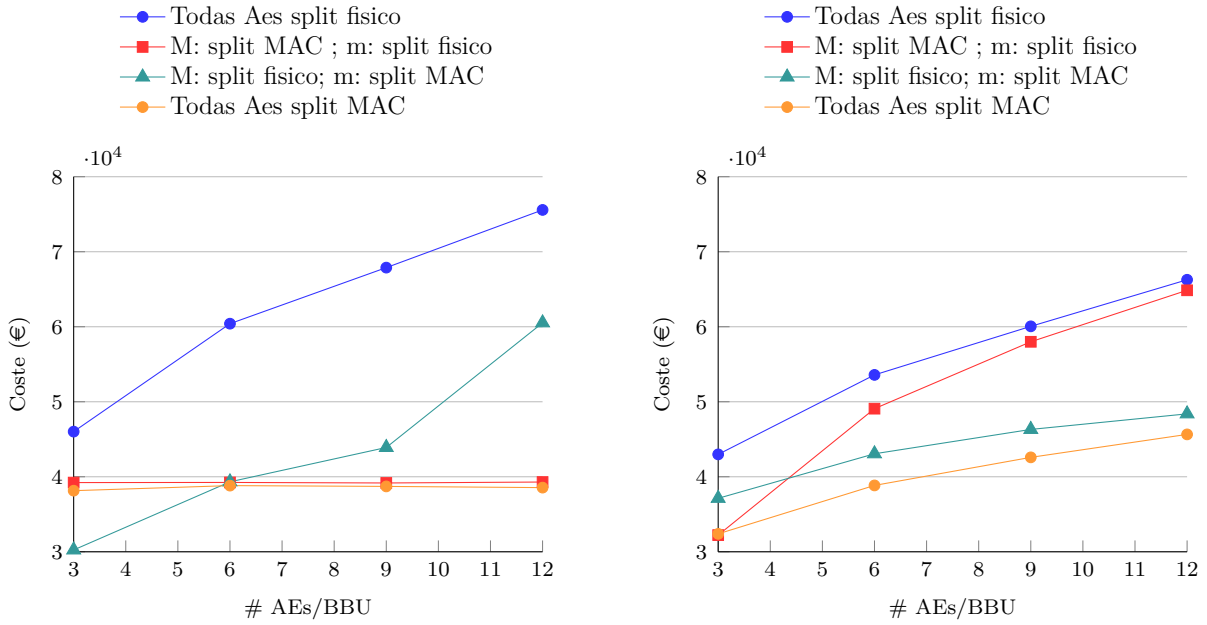
Para un split de tipo físico, los costes de despliegue son irrelevantes, ya que siempre se usa solución vía fibra óptica. En el caso de tener un split de tipo MAC, la solución que más interesa utilizar dependerá de la distancia del enlace (siendo mejor para la solución de fibra siempre que la distancia de enlace sea inferior a 280 y 400m según sea Macro o Micro, respectivamente).

Una vez se ha analizado el impacto que introduce en el algoritmo la introducción de las tecnologías en la función de coste, se presentan los resultados obtenidos en las Figuras 4.15 y 4.16, que se corresponden, respectivamente, con el coste medio de un enlace y el porcentaje de conexiones. Se distingue entre tipos de estación base y número de BS que se pueden conectar a la estación en cada caso, para un número fijo de BBUs, concretamente, 10.

Se puede ver que en el caso de la función de coste, el comportamiento con split MAC es bastante similar en ambos tipos de estaciones base. Si sólo estuviese limitado a tener la tecnología inalámbrica, su comportamiento sería plano, pero se reduce y con tendencia ascendente por el uso de fibra en los enlaces cortos. Por otro lado, en el caso del split físico el coste crece notablemente al aumentar el número de Aes por BBU, debido a que hay un mayor número de conexiones, como se ve en la gráfica correspondiente, en la que la conectividad llega a ser del 100 %

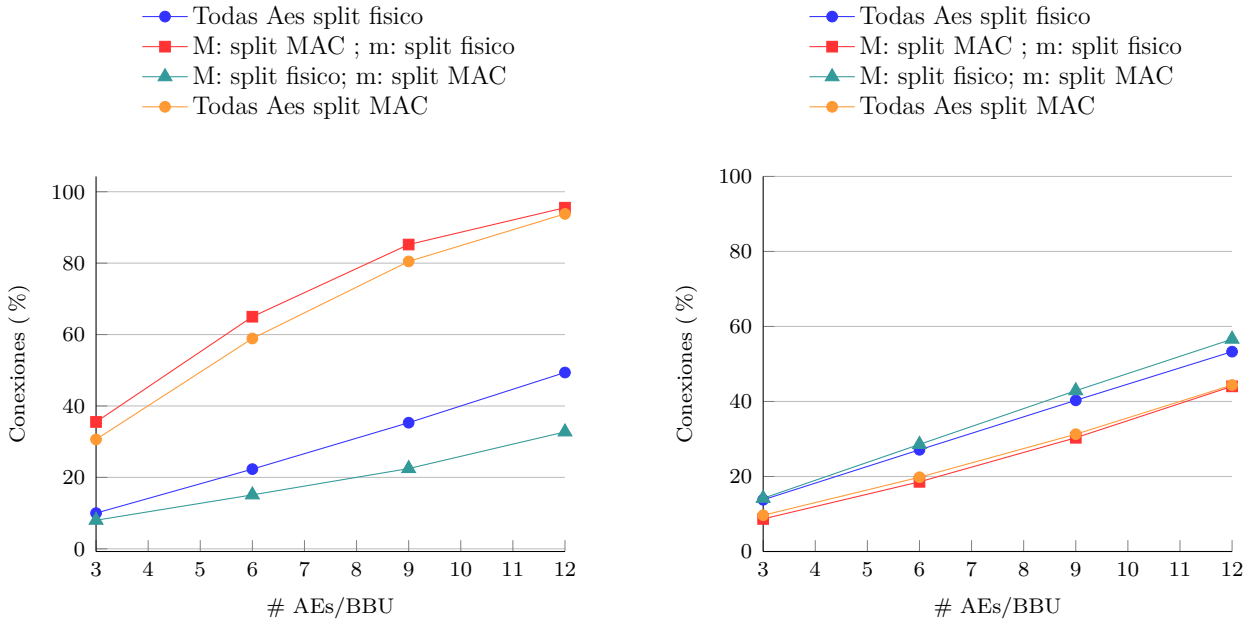
Una vez se ha llegado a este punto del trabajo, se ha encontrado la solución para la posición de las BBUs dentro del escenario, de modo que minimice el coste. Todo ello, conociendo a priori el tipo de split que se le aplica a cada estación base, y aceptando que se van a poder conectar a una BBU, sea cual sea el split de la estación base donde se localiza.

El objetivo en la segunda parte del trabajo consiste en llegar a la solución óptima a través de programación basada en restricciones, partiendo de un problema en el que no se conozca el split de ninguna estación base y limitando a que a una BBU sólo se puedan conectar estaciones que compartan su mismo split, que tampoco es conocido a priori.



(a) Análisis de las Macros (b) Análisis de las Micros

Figura 4.15: Coste medio de enlace para 10BBUs según el tamaño y el split de las estaciones base



(a) Análisis de las Macros (b) Análisis de las Micros

Figura 4.16: Porcentaje de conexiones para 10BBUs según el tamaño y el split de las estaciones base

Capítulo 5

Selección óptima de splits a partir del uso de programación con restricciones

5.1. Programación con restricciones

En esta parte del trabajo el objetivo principal es tratar de optimizar el problema planteado en el capítulo anterior, minimizando el coste del despliegue de una red, pero en este caso haciendo uso de una solución basada en programación con restricciones [24] y [25].

5.1.1. ¿Qué es un problema basado en restricciones?

Son un tipo de problemas que se resuelven a través de resolución de ciertas restricciones, que no son diferentes a las que se podrían encontrar en el mundo real, tales como no sobrepasar un presupuesto límite, por ejemplo.

Cuando estos problemas tienen una gran cantidad de variables y restricciones, la capacidad de los seres humanos para resolverlos es limitada, por lo cual se debe recurrir a los ordenadores, y más concretamente, a los Constraint Satisfaction Problems (Problemas de satisfacción por restricciones) (CSPs). Algunos problemas NP-Completos pueden pasar a ser polinomiales, utilizando los CSPs, que convierten el grafo de conectividad en un árbol.

Estos problemas, se pueden utilizar sobre cualquier lenguaje, pero si fuera orientado a objetos facilitaría la tarea, de ahí que la utilización de C++ como lenguaje de programación sea una elección idónea.

Como la mayor parte de problemas de inteligencia artificial, los CSPs se resuelven mediante búsqueda. Lo que les hace únicos respecto al resto de estos problemas es que hay una estructura estándar, permitiendo de esta manera resolver el problema sin

conocerlo en profundidad. Además, son conmutativos, por lo que ante cualquier orden en la búsqueda, darán la misma solución.

Algunas de las aplicaciones prácticas de este tipo de problemas son:

- Realizar el scheduling y el timetabling
- Bioinformática, para secuenciar el ADN
- Ingeniería eléctrica, para hacer el layout
- Telecomunicaciones, para las siguientes generaciones de tecnología
- Scheduling de satélite

5.1.2. Definición formal de CSPs

Cualquier problema de este tipo, se basa en una serie de variables que pueden tener valores dentro de un determinado dominio (siendo estos valores enteros, en la mayoría de los casos). Los valores que se asignen definitivamente a cada variable han de cumplir además una serie de restricciones impuestas, para cada variable, o relaciones entre las mismas.

Un problema basado en restricciones puede tener múltiples soluciones, una o ninguna. Para hacer que prevalezca una de esas soluciones se pueden utilizar restricciones de preferencia, o asignar un coste a cada solución, tratando así de maximizar o minimizar el mismo, dependiendo de la naturaleza del problema a resolver. En el caso de este trabajo de Fin de Máster, se tratará de minimizar una función de coste, para encontrar la solución óptima de split, partiendo de la solución que da el algoritmo del Capítulo anterior, es decir, conociendo ya, cuáles son las estaciones base óptimas para la colocación de las BBUs en base a la función de coste que minimiza la distancia.

5.1.3. Restricciones

Es una de las partes más importantes en la resolución de CSPs, puesto que en función del modelado de las restricciones, la librería que se utilice para la resolución de este tipo de problemas será más o menos eficiente. Por todo esto, es muy importante encontrar un modelado apropiado de estas restricciones, que pueden ser:

- n-arias: si engloban a n variables (también denominadas, restricciones globales)
- unarias: si engloban sólo una variable
- binarias: si engloban a dos variables

Las restricciones de tipo n-ario se pueden reducir a restricciones del tipo binario, pero no siempre es una buena idea, porque puede causar el crecimiento exponencial de este tipo de restricciones, haciendo que el modelo sea intratable computacionalmente.

La restricción global más utilizada es, AllDifferent (que fuerza a que todas las variables tengan asignados valores diferentes). Esta restricción se puede descomponer fácilmente en muchas restricciones NotEqual entre las variables.

Además, las restricciones ayudan a la representación del problema en forma de grafo, donde se tienen nodos, que son las variables; y enlaces, que representan la restricción que hay entre las posibles variables.

5.1.4. Variables

Como se ha dicho con anterioridad, cada una de las variables a las que hay que asignar un valor en el problema, tienen un dominio que se va limitando, con la propagación de las restricciones entre las diferentes soluciones que se van obteniendo.

Es muy importante esta propagación, que puede hacer que el algoritmo sea mucho más eficiente. Para lograr esa mejora, hay que fijarse en aspectos tales como el orden de asignación de las variables. Por ejemplo, existe un algoritmo (Minimum Remaining Values (MRV)) que escoge la variable a la que menos valores de su dominio le quedan aún disponibles. De este modo, se puede conocer rápidamente una rama de árbol "muerta", que será el punto donde una variable no tenga valores disponibles en el dominio, porque se han ido eliminando a medida que no cumplían las restricciones impuestas. Esa solución no sería, por tanto, válida, permitiendo reducir el árbol de búsqueda.

Hay múltiples algoritmos que funcionan dinámicamente, a medida que la búsqueda avanza. Así, se reduce la cardinalidad del dominio (para ello, hay alternativas que eliminan los valores que no cumplen restricciones), pero en el caso de nuestro problema, como queremos encontrar la solución óptima, se va a tratar de hacer un barrido sobre todas las posibles asignaciones, por lo que no se va a reducir el árbol de búsqueda, aunque sí encontrar ramas por donde no se pueda continuar, gracias a las restricciones impuestas, respecto al dominio original.

Una vez seleccionada la variable a la que se le va a asignar el valor, hay que fijarle. Un criterio lógico lo proporciona el algoritmo Least Constraining Value (LCV), cuyo principio se basa en elegir el valor que elimina menos opciones del dominio de las demás variables.

Estas dos acciones que acabamos de presentar, son las que forman el algoritmo de búsqueda, elegir cada variable y el valor de la misma. En una búsqueda en árbol, cada variable es un nodo y cada rama que sale del nodo son los valores del dominio que puede tomar. La profundidad del árbol, si tenemos n variables será de n.

Cuando se llega a un nodo del que no pueden salir más ramas, indica que todo su dominio no cumple con las restricciones, por lo que por ese camino no se puede llegar a una solución válida. Por lo cual, hay que volver hacia atrás, utilizando el algoritmo Back Tracking, que genera el árbol a medida que la búsqueda progresa.

Respecto a la propagación de restricciones, cuando se elige un valor para una variable hay que ver si es consistente con las restricciones de nuestro problema. Para realizar esta operación, existe el algoritmo Forward Checking, que actualiza temporalmente los dominios de las variables para que sean consistentes con las restricciones, y con los valores de las variables ya asignadas.

De esta manera, el dominio de alguna variable puede quedar vacío, y habrá que asignar otro valor a la variable actual, intentando de este modo no llegar a un "dead end". Si no hay más valores que asignar a esta variable, habrá que volver hacia atrás, Back Tracking. Esta opción, que contempla el uso de Forward Checking combinado con Back Tracking, es más eficiente que utilizar sólo el segundo de estos algoritmos.

5.1.5. Consistencia local

La asignación de un valor a una variable es consistente cuando cumple las restricciones actuales y aquellas que se generan en ese momento. Los problemas K-consistentes son aquellos en los que nunca hay que volver atrás, ya que la asignación consistente de una variable implica la asignación consistente de las demás. Para chequear la consistencia local se utilizan los siguientes algoritmos:

- Arc Consistency: examina la consistencia del valor asignado entre dos variables. El chequeo es en ambas direcciones, por lo que es más fuerte que el Forward Checking. Se llama al algoritmo tras cada instanciación de variable, para que elimine del dominio todos los valores inconsistentes para la asignación actual.
- Path Consistency: chequea tríos de variables, en vez de pares como el Arc Consistency, por lo que tiene una mayor complejidad.
- Bound Consistency: examina la consistencia del valor máximo y mínimo del dominio, eliminando así parte de la búsqueda, porque si el máximo y mínimo no cumplen la restricción, todos los demás valores tampoco lo harían; así se evita ir chequeando uno a uno todos ellos. Este tipo de chequeo de consistencia no es válido para el problema que se estudia.

5.1.6. Simetría

Una manera de evitar realizar una búsqueda de todas las soluciones del problema es aprovechar simetría entre las mismas. Si se encuentra fácilmente entre las asignaciones, implica que éstas podrían ser intercambiables, con lo que se podrían encontrar soluciones nuevas al problema de manera sencilla.

Encontrar dichas simetrías, permite reducir el tiempo de búsqueda, ya que si se localiza con las restricciones del modelo, una asignación no factible, todas las simétricas tampoco lo serán y viceversa.

5.1.7. COP

Son el tipo de problemas al que se hace frente en este trabajo, y tienen que ser optimizados en base a una función dada, representada en forma de restricción.

Cuando se llega a una solución, hay que asociarla con un 'ranking' dentro de la función objetivo, calculando así todas las soluciones y quedándonos con la mejor dentro de dicha satisfacción. De este modo, se habrá minimizado/maximizado la función que define el problema.

5.2. Librería Gecode

Para resolver un problema como los que se han presentado en el apartado anterior, se ha de tener acceso a una librería que proporcione todas esas funcionalidades. En este caso, se ha decidido utilizar la librería Gecode [26].

Esta librería ofrece la capacidad de modelar las restricciones, hacer el branching, esto es, conocer el dominio disponible para cada una de ellas tras imponer las restricciones, y utilizar variables simples y complejas muy intuitivamente, a la vez que da la posibilidad de utilizar diferentes métodos de búsqueda.

5.2.1. Arquitectura de la librería Gecode

A continuación se va a explicar las partes en las que consiste la arquitectura de esta librería, que facilita que el usuario, sólo se encargue de implementar el modelado del problema, y mediante llamadas transparentes al kernel, resolver fácilmente el mismo.

Como se observa en la Figura 5.1, el kernel de la librería ofrece las funcionalidades y el soporte donde están construidos, tanto los tipos de variables (set, float e int), como todas las operaciones referentes a las restricciones, y a las máquinas de búsqueda.

5.2.2. Algoritmo creado para la optimización

En esta parte de la memoria se va a discutir como se ha modelado el problema utilizando una combinación de la librería Gecode, con las que incorpora C++ de manera estándar. Concretamente se ha estructurado de la siguiente manera: por un lado, se tiene un main principal, en la que se llamará a la clase, donde se utilizan las herramientas que proporciona Gecode.

- Main : en el main del programa se hacen las tareas de procesar los resultados obtenidos al ejecutar, para un escenario concreto, el algoritmo explicado en el Capítulo anterior, ya que se van a utilizar como variables de entrada al algoritmo actual.

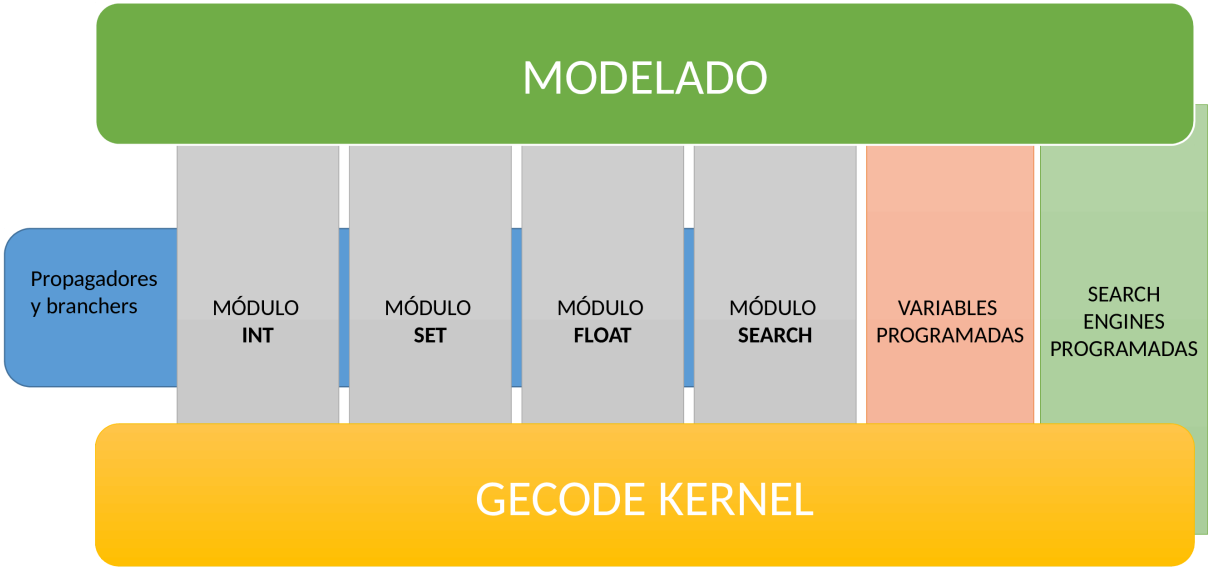


Figura 5.1: Arquitectura de Gecode

Es decir, si antes la incógnita era: ¿cuáles son las estaciones base que tienen que ser BBUs para minimizar la función de coste?, ahora es: Una vez conocidas esas estaciones base óptimas para situar las BBUs, ¿cuál es el split óptimo de todas las estaciones base para esa configuración de red?. Para explicar el funcionamiento, se presenta el Código 5.1, que se comentará a continuación.

```
// MAIN FUNCTION
    int main ( int arg c, char* argv[]
     {
        // CREATE MODEL AND SEARCH ENGINE
        Options opt("\n␣P-Mediana");
        opt.solutions(0);
        opt.parse(argc, argv);

        // SEARCH AND PRINT ALL SOLUTIONS
        IntMinimizeScript::run<ConstraintProgr, BAB, Options>(opt);

        //Export the best solution to a file
        return(0);
    }
```

Código 5.1: Descripción del Main

Descripción del Código 5.1:

- Crear el modelo y la máquina de búsqueda : para ello se deben seleccionar las opciones con las que se quiere configurar la búsqueda. Esto puede ser, desde el número de soluciones que se desean (en este caso, todas las posibles, ya que se quiere la mejor, por lo que hace falta explorar todas), si se pretenden exportar a un archivo de texto (si aumenta mucho el número de soluciones

encontradas, el archivo crecería exponencialmente, por lo que sólo se registra la solución escogida como la mejor). Además, también se selecciona el tipo de búsqueda que se realiza, (en este caso, una búsqueda basada en Branch and Bound (BAB), la cual es la más utilizada cuando se quiere encontrar todas las soluciones posibles a un problema, también existe Depth First Search (DFS) o Limited Discrepancy Search (LDS) que son más apropiadas, cuando se quiere encontrar una solución válida, sin preocuparse de minimizar o maximizar un problema).

- Buscar y pintar las soluciones : en este caso sólo se pinta la solución que minimiza el problema.

- Clase que implementa el algoritmo con Gecode: nos vamos a basar en el siguiente esquema 5.2 del código para explicar como se ha estructurado el algoritmo

```
#include <gecode/driver.hh>
#include <gecode/int.hh>
#include <gecode/float.hh>
#include <gecode/set.hh>
#include <gecode/minimodel.hh>

using namespace Gecode;
using namespace std;

class ConstraintProgr : public IntMinimizeScript
 {
    public:

    IntVarArray q;
    ConstraintProgr(const Options& opt) : IntMinimizeScript(opt) {

        --> RESTRICCIONES: Relaciones entre variables del array q
            -Las posiciones donde se encuentren las BBUs van a valer 1 o 2
             dependiendo del split, fisico o Mac
            -El resto de las estaciones tendran el valor de la BBU a la que
            se conecten, no pudiendo ser ese valor 1 o 2 porque es el que
            se corresponde con los splits

        --> BRANCHING : indicando a la q que se empiece a seleccionar la
        variable que menos dominio disponible tenga y el valor de preferencia
        sea el minimo disponible

        --> WAIT: una vez asignado valores a todo el array q, comprobar que
        la solucion es consistente, respecto a las distancias maximas de las
         tecnologias que definen los splits

        --> COSTE MINIMO : una vez sabido que la solucion es consistente,
        se obtiene el coste, y se va guardando la solucion que da el coste minimo.
    }

    --> SEARCH SUPPORT

    --> PRINT SOLUTION
};

--> MAIN FUNCTION
```

Código 5.2: Descripción del algoritmo

- Variables: todo el desarrollo del algoritmo se va a basar en dar diferentes valores a un array de variables, que va a indicar, según la posición en el array:

  - Posición en el array de una BBU: indicará el tipo de split (Físico o MAC) configurado en esa estación base, en la que se colocará una BBU
  - Posición en el array de una estación base normal: indicará a qué BBU está conectada, con lo cuál esa estación base tendrá el mismo split que la BBU indicada.
- Restricciones: Se impondrán diferentes restricciones para ese array de variables, para completar el dominio de cada una de ellas, según el tipo: si es BBU, o no.
- Branching: Se indica el orden en el que se va a realizar la búsqueda, tanto en orden de asignación de variables dentro del array. Primero se asignarán las que menos valores tienen en el dominio, lo que disminuye el tiempo de búsqueda de 'ramas muertas', en caso de buscar una única solución; y a continuación, el orden de asignación de valores dentro del dominio de la variable seleccionada, en este caso de menor a mayor, por poner un criterio (Código 5.3).
- Wait: Este Wait del proceso de Linux se utiliza, porque hay restricciones que dependen del valor que se haya dado a las variables tras la asignación, por lo que no se puede imponer como restricción previa. Este Wait se llama, tras cada asignación del array solución, chequeando que las asignaciones son correctas, según el split y la conexión que se haya impuesto en la asignación. Ya que esos splits están asociados a diferentes tecnologías (ondas radio o fibra óptica) que tienen un alcance limitado. En caso de no cumplirse los límites físicos de las tecnologías con la solución encontrada, se marca como fallo y no se contabiliza como solución.
- Coste: Por último, una vez que se ha pasado el filtro con el Wait, si la asignación es aceptada, se calcula el coste que conllevaría la utilización de esas tecnologías para los splits asignados, según la distancia entre las estaciones base del escenario.
- Clonar el espacio: La búsqueda en Gecode, se basa en volver a computar una solución a partir de un clon del espacio creado como solución anterior (Código 5.4).

```
branch(*this, q, INT_VAR_SIZE_MIN(), INT_VAL_SPLIT_MIN());
```

Código 5.3: Descripción del branching

```
/// Constructor for cloning \a s
ConstraintProgr(bool share, ConstraintProgr& s) : IntMinimizeScript(share, s) {
    q.update(*this, share, s.q);
}

/// Perform copying during cloning
virtual Space* copy(bool share) {
    return new ConstraintProgr(share, *this);
}
```

Código 5.4: Descripción del clonado

5.3. Análisis de los resultados

Se utiliza un escenario con las siguientes características:

- Dimensiones : 1500m x 1500m
- Estaciones base :
  - Macro estaciones base : 7 en total
  - Micro estaciones base : 7 en total
- Configuraciones : se va a utilizar la configuración que se determinó con mejor comportamiento en el capítulo anterior, es decir, la BBU puede estar en cualquier estación base, sea Macro o Micro

  El despliegue de las estaciones base de tipo Macro se configura en dos anillos regulares y el despliegue de las micro es completamente aleatorio (Figura 5.2)

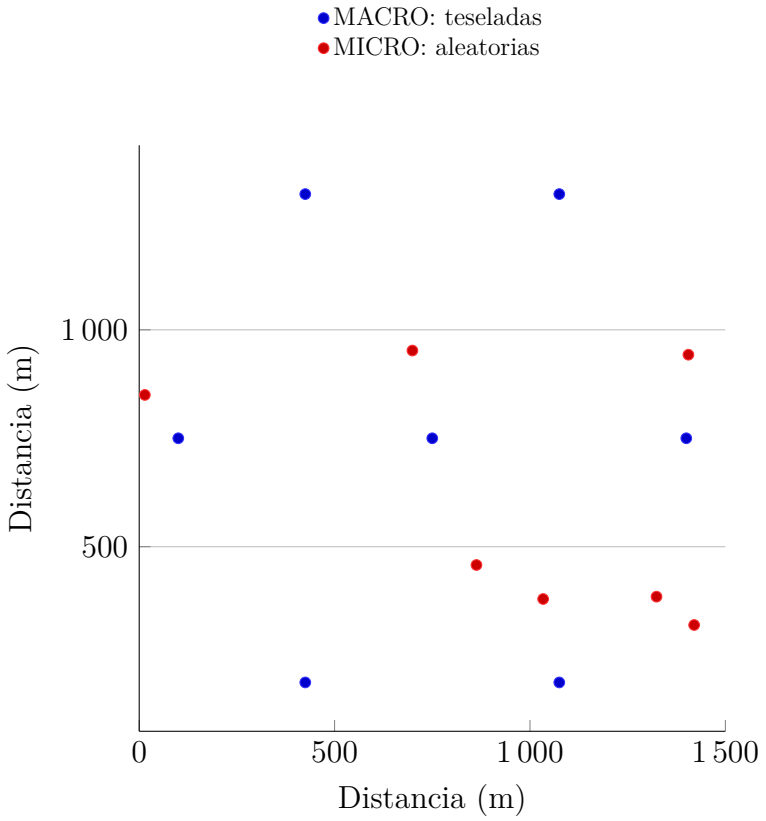


Figura 5.2: Ejemplo de despliegue del escenario

A continuación, se va a presentar una comparativa de los resultados obtenidos utilizando una configuración de split fija (algoritmo, basado en la solución greedy, presentado en el Capítulo anterior) y usando el algoritmo realizado para analizar el comportamiento de programación basada en restricciones.

Concretamente, las configuraciones de split fijas serán :

- Macros con split físico y Micros con split físico, usando fibra como tecnología en los enlaces para ambos casos
- Macros con split Mac y Micros con split físico, usando microondas en las Macros y fibra en las Micros
- Macros con split físico y Micros con split Mac, usando fibra óptica en las Macros y ondas milimétricas en las Micros
- Macros con split Mac y Micros con split Mac, usando microondas y ondas milimétricas como tecnologías, respectivamente

De este modo, se va a realizar la comparativa de las cuatro configuraciones fijas, y la utilización del Constraint Programming. Para poder ejecutar el algoritmo que analice la solución de split óptima a partir de la utilización del Constraint Programming, es necesario ejecutar previamente el algoritmo del capítulo anterior, eligiendo así las BBUs que minimizan el coste, según la distancia de enlace.

Es decir, con este nuevo algoritmo, ya se tienen seleccionadas las estaciones base que serán BBUs utilizando el algoritmo greedy y las posiciones de las mismas; y lo que se fijará será el split óptimo para tratar de reducir aún más el coste. Es importante recordar que el coste de las tecnologías es el que se indica en la Figura 4.14.

En la Figura 5.3 se puede observar el comportamiento del coste en función del número de BBUs. En todas las configuraciones, ya sea las de la primera fase, usando sólo el algoritmo greedy (en azul) como con la utilización de programación con restricciones (en rojo), el coste disminuye a medida que aumenta el número de BBUs, ya que o bien depende de la distancia, en el momento que se utilizan configuraciones en las que se tiene fibra, o bien se elimina el coste que supondría la nueva BBU, si fuese una estación base normal.

El coste óptimo viene dado por los resultados con programación con restricciones, ya que examina todas las posibilidades de configuración. Se ve que para este escenario, cuando el número de BBUs es menor a cuatro, cualquiera de las configuraciones fijas, está bastante alejadas de la óptima. En cambio, al tener más de cuatro BBUs, la configuración que siempre se selecciona como la óptima, es la utilización de Fibra óptica en las Micro-estaciones base, y microondas en las Macro-estaciones base.

Por lo cual, se concluye que para este escenario, una vez se tengan más de cuatro estaciones base, no parecería justificable la utilización de programación con restricciones ya que dará el mismo resultado que utilizar directamente Fibra óptica en las Micro-estaciones base, y de microondas en las Macro-estaciones base. De este modo, se ahorraría mucho tiempo (Tabla 5.1), desestimando cómputo innecesario, ya que el rendimiento del algoritmo de programación de restricciones disminuye en gran medida en cuanto se

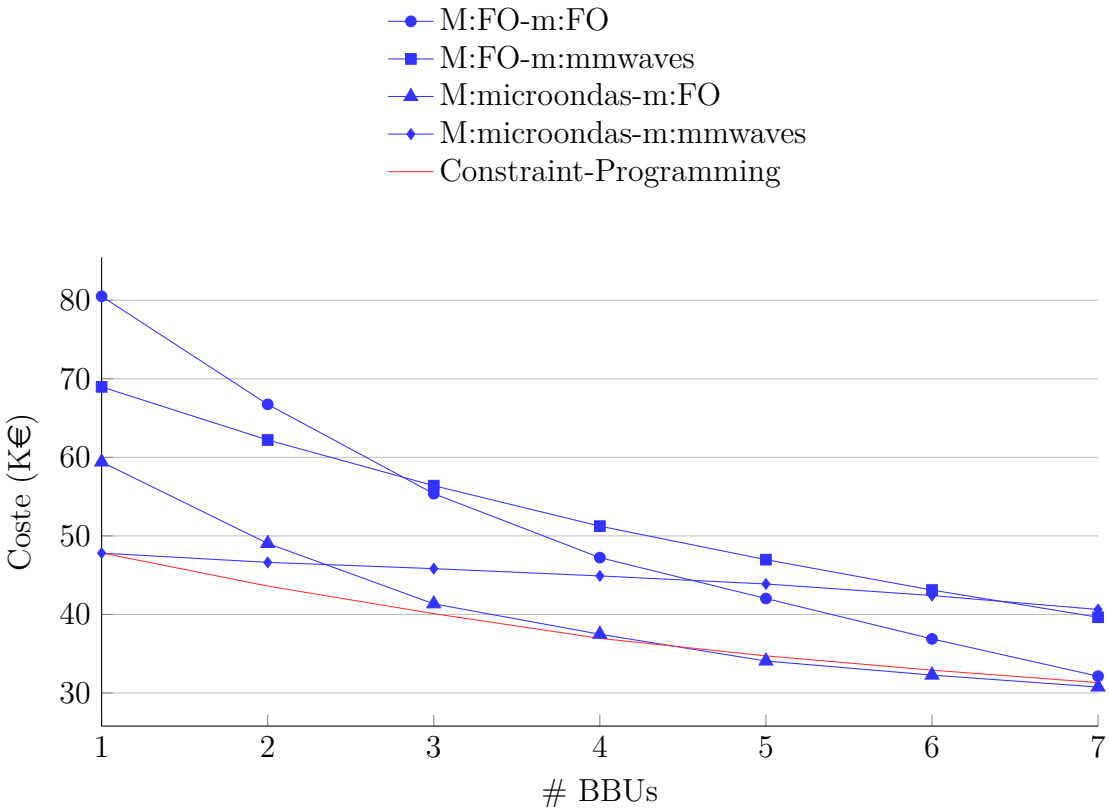


Figura 5.3: Análisis del coste requerido con ambos algoritmos

aumenta el número de BBUs, pues implican una gran cantidad de combinaciones, y el tiempo de búsqueda crece exponencialmente (Tabla 5.1).

Pero, hay que indicar que la comparativa entre ambas opciones; split fijo y optimización mediante el uso de programación con restricciones, no es del todo justa. Esto es debido a que en la programación con restricciones se está contemplando una condición más: que todas las estaciones base que se conecten a una BBU tienen el mismo Split. Por ello, para opciones donde hay más controladores, son ligeramente mejores los resultados donde no se realiza optimización, ya que no considera esa restricción.

Tabla 5.1: Ejemplo de ejecución del algoritmo con Constraint Programming para diferente número de BBUs

#BBUs	**Tiempo de cómputo (mseg)**
1	1.389
2	19.579
3	1507.456
4	15417.128
5	73153.107
6	124333.755
7	134457.871

En segundo lugar, se presenta la Figura 5.4, esta vez sólo representando el análisis del Constraint Programming, donde se indica la distribución de los splits en las estaciones base presentes en el escenario, según el número de BBUs que se tengan. Se ve claramente que al principio, siempre se elige Split de tipo Mac, porque las estaciones base están tan alejadas como para que la fibra no sea razonable. Pero esa tendencia cambia a medida que aumenta el número de BBUs en el escenario, porque inevitablemente la distancia a la BBU de cada estación base se reduce considerablemente, hasta que a partir de cinco BBUs en el escenario, son más las estaciones base, que utilizan un split de capa física y, por lo tanto, fibra óptica como tecnología.

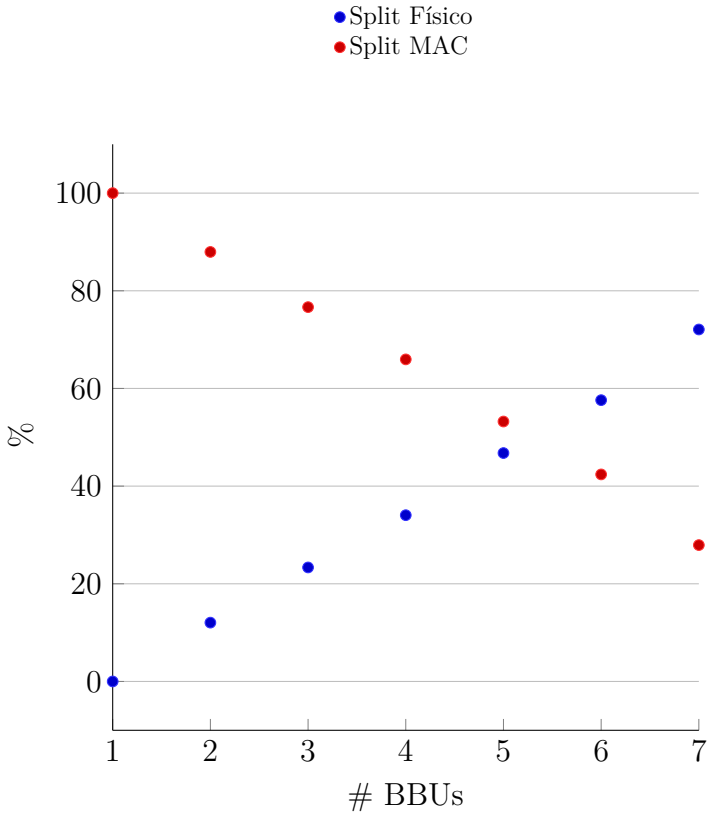


Figura 5.4: Distribución de los splits en las estaciones base del escenario

A continuación, se presenta en las Figuras 5.5 y 5.6 la distribución de estaciones base en cada BBU. Se puede ver que es más justa con cajas más estrechas, indicando una dispersión menor de los resultados a medida que hay más unidades de banda base en el escenario. Por otro lado, teniendo un escenario tan pequeño como el que se presenta, se deduce además que para 7 BBUs, cuando se analiza el Constraint Programming, no se tienen estaciones base conectadas a alguna de las BBUs, por lo que no tiene sentido un escenario con tantas BBUs para pocas estaciones base. Sin embargo, permite percibir que, el tiempo de cómputo, crece exponencialmente, teniendo ante nosotros un problema intratable, para la capacidad de cómputo del servidor en que se ejecuta el problema.

El comportamiento de la función de distribución es análogo, tanto en las configuraciones donde se fija el split, como en las que se elige inteligentemente, minimizando el coste total del despliegue, salvo por el hecho de que en las configuraciones con el split fijo para los diferentes tipos de estación base, no llega a aparecer ningún controlador sin conexiones. Hay que tener en cuenta que al analizar el comportamiento del escenario con un 'split fijo', simplemente se elige el tipo de tecnología que se va a utilizar según el tipo de estación base de que se trate, pero el controlador puede estar teniendo un split y los elementos de acceso conectados a él, otro, por lo cual no es estrictamente ecuánime la comparativa.

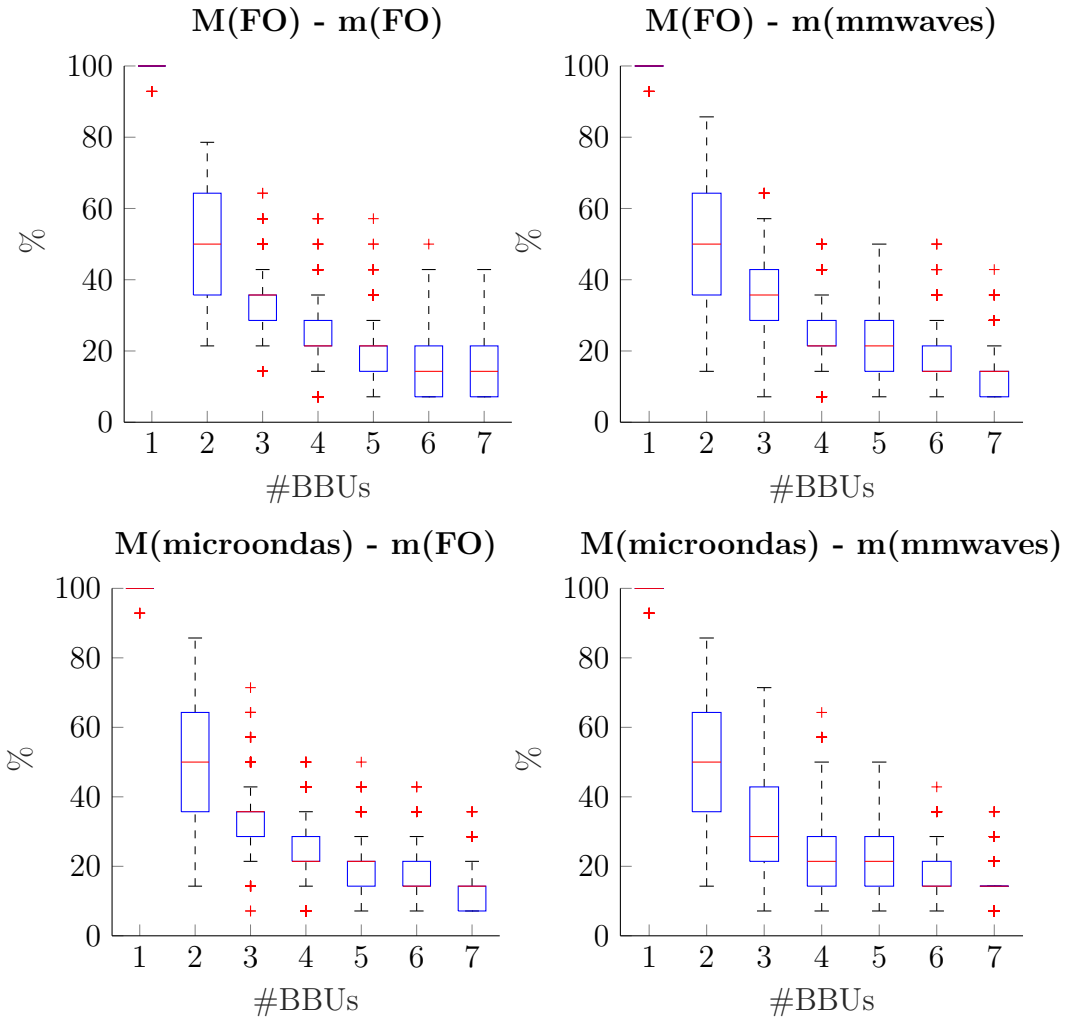


Figura 5.5: Distribución de las estaciones base en las distintas BBUs cuando se utiliza el algoritmo greedy fijando los splits

Por último, se va a realizar un análisis del rendimiento del algoritmo (Tabla 5.2) que analiza la utilización de la librería Gecode, donde se ve el crecimiento exponencial de soluciones, fallos y propagaciones para una ejecución del algoritmo basado en

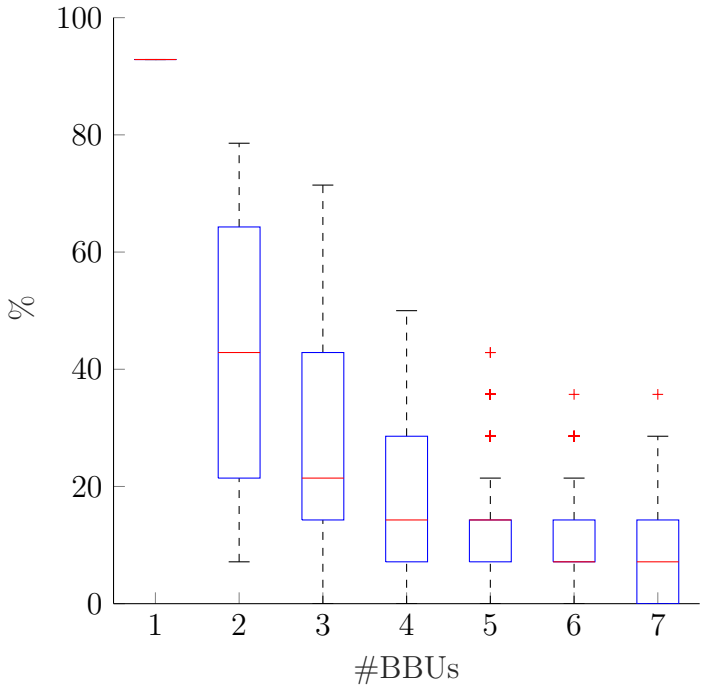


Figura 5.6: Distribución de la carga de las estaciones base en las distintas BBUs cuando se utiliza el algoritmo de programación con restricciones

programación con restricciones, según el número de controladores que se fijan por configuración en el escenario.

Tabla 5.2: Rendimiento del algoritmo con Constraint Programming

#BBUs	**Soluciones**	**Fallos**	**Propagaciones**
1	2	0	10
2	10240	6144	24593
3	43092	1374084	4964119
4	325432	2078818	4946774
5	1444608	15332608	30277070
6	9649920	52850080	288306986
7	14054400	91359104	783220191

Capítulo 6

Conclusiones

Mediante este trabajo se ha querido realizar un estudio de viabilidad, de la posible implementación de una arquitectura C-RAN, en los futuros sistemas 5G, sobre un entorno C++.

Para realizar el análisis planteado se han tenido que implementar dos algoritmos diferentes:

- Algoritmo para resolver una P-Mediana basado en soluciones heurísticas, concretamente, en una solución greedy
- Algoritmo que se basa en programación con restricciones para alcanzar una solución óptima de funcionalidades viables a llevar a la nube

Cada una de las dos fases de estudio de este trabajo se basan precisamente en los dos algoritmos mencionados con anterioridad. En la primera fase sólo se hace uso del algoritmo de resolución de la P-Mediana, mientras que en la segunda fase se emplean ambos.

Entre los resultados obtenidos, se puede destacar la posibilidad de escoger la configuración que mejor se comporta para este tipo de problemas: en este caso es la que posibilita a todos los elementos de acceso actuar como controladores, lo que conlleva un aumento del tiempo de cómputo en los algoritmos. Por otro lado, se concluye que el comportamiento de la longitud media de enlace necesaria es *siempre* decreciente, con el incremento de controladores dentro del escenario, siempre que no se limite el número de estaciones que se pueden conectar a cada controlador. Además, también existe la posibilidad de desconectar del backhaul de la red algunos elementos de acceso, según sea la situación de los mismos en el escenario, las tecnologías de conexionado, o los límites de ancho de banda que pueden gestionar los controladores.

En cuanto al algoritmo que consigue la optimización de las funciones de red que se llevan a la nube, hay que ir a los resultados que se plantean utilizando programación con restricciones, ya que examina todas las posibilidades de splitting. En el escenario que se analizó, se desplegaban 14 estaciones base, debido a la alta carga computacional del

problema en cuanto crecía el número de soluciones a explorar. Se concluye que cuando el número de BBUs es menor a cuatro, cualquiera de las configuraciones fijas está bastante lejos de la óptima. En cambio, al tener más de cuatro BBUs, la configuración que siempre se seleccionaría a priori como la óptima, es la que utilizan Fibra óptica en las Micro-estaciones base, y microondas en las Macro-estaciones base. No es una comparativa completamente justa, ya que en la programación basada en restricciones se tiene en cuenta la limitación de que sólo las estaciones base con el mismo split que la BBU, pueden conectarse a ésta.

Por lo cual, para el escenario analizado, una vez se tengan más de cuatro estaciones base, la utilización de programación con restricciones no se justificaría, en caso de que diese igual que las estaciones base que se conectan a una BBU determinada tengan splits distintos a ésta, ya que se obtiene el mismo resultado que utilizar directamente Fibra óptica en las Micro-estaciones base, y de microondas en las Macro-estaciones base, con el correspondiente ahorro en tiempo de cómputo.

Después del estudio realizado en este trabajo, se concluye que este tipo de arquitecturas podrían aplicarse en sistemas del mundo real en un futuro no muy lejano. Aún así, se considera el C-RAN como una arquitectura con una madurez insuficiente como para implementarse en la actualidad de una manera estandarizada, ya que se requiere de total flexibilidad de los controladores, para decidir, según el escenario en el que estén implementados, utilizar uno y otro nivel de splitting. Habría que realizar un análisis específico de cada escenario concreto para conocer, como en el ejemplo de este trabajo, cuál es la elección óptima de funcionalidades de red que pasan a la nube.

Por último, es importante remarcar que se han cumplido todos los objetivos marcados al inicio del Trabajo, donde se requería:

- Crear una herramienta de simulación de redes LTE.
- Implementar un algoritmo que resuelva una P-mediana
- Implementar un algoritmo que encuentre la división de funciones óptima para un escenario concreto optimizando con programación a partir de restricciones.
- Combinar todos estos ingredientes, para encontrar una localización óptima en una red de comunicaciones de los controladores, que proporcionarán la inteligencia a las estaciones base, y el nivel de splitting óptimo que se puede utilizar en cada controlador, para un escenario concreto.

6.1. Líneas futuras

A raíz del estudio desarrollado a lo largo de este trabajo, surgen líneas futuras de desarrollo. Como función adicional que podría ser añadida a la herramienta destaca un algoritmo que haga uso todo lo que se ha presentado con anterioridad, y que realice la búsqueda del split óptimo para todas las estaciones base, sin partir de la solución

basada en la P-Mediana. Para ello, el algoritmo de programación con restricciones no partiría de una solución dada, sino que tendría que optimizar también este parámetro.

Junto con la optimización completa del escenario planteado, se podrían analizar diferentes configuraciones de red, con otro número de estaciones base, en los que se pueda considerar interesante la utilización de estas técnicas.

Además, también sería interesante incluir en el primer algoritmo la restricción de que sólo las estaciones base con el mismo nivel de splitting que los controladores, puedan conectarse a éstos, de modo que la comparativa de ambos algoritmos fuera lo más justa posible.

Bibliografía

[1] ITU. IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond, M Series, Recommendation ITU-R M.2083-0. 0:21, 2015.

[2] White Paper. Architecture for 5G. pages 1–24.

[3] John Naylon. Managing OPEX and CAPEX in small cell backhaul. (June), 2013.

[4] Christian Hoymann, David Astely, Magnus Stattin, Gustav Wikstrm, Jung Fu Thomas Cheng, Andreas Hglund, Mattias Frenne, Ricardo Blasco, Joerg Huschke, and Fredrik Gunnarsson. LTE release 14 outlook. *IEEE Communications Magazine*, 54(6):44–49, 2016.

[5] L. Valcarenghi, K. Kondepu, F. Giannone, and P. Castoldi. Requirements for 5G fronthaul. *International Conference on Transparent Optical Networks*, 2016-Augus:1–5, 2016.

[6] Mona Jaber, Muhammad Imran, Rahim Tafazolli, and Anvar Tukmanov. 5G Backhaul Challenges and Emerging Research - A survey. *IEEE Access*, 3536(c):1–1, 2016.

[7] Paula Rodríguez, Paula Sarasúa, Luis Diez, and Ramón Agüero. Simulación genérica a nivel de sistema para soluciones avanzadas de gestión de recursos. (Jitel):27–29, 2017.

[8] Christian Mehlführer, Josep Colom Ikuno, Michal Šimko, Stefan Schwarz, Martin Wrulich, and Markus Rupp. The Vienna LTE simulators - Enabling reproducibility in wireless communications research. *EURASIP Journal on Advances in Signal Processing*, 2011(1):29, 2011.

[9] NS-3. ns-3 Manual. 2016.

[10] Giuseppe Piro, Nicola Baldo, and Marco Miozzo. An lte module for the ns-3 network simulator. pages 415–422, 2011.

[11] C. Schneider and R. S. Thomä. Evaluation of lte link-level performance with closed loop spatial multiplexing in a realistic urban macro environment. pages 2725–2729, March 2012.

[12] R. M. Fujimoto, K. Perumalla, A. Park, H. Wu, M. H. Ammar, and G. F. Riley. Large-scale network simulation: how big? how fast? pages 116–123, Oct 2003.

[13] Joshua Pelkey and George Riley. Distributed simulation with mpi in ns-3. pages 410–414, 2011.

[14] Josep Colom Ikuno. Lte link- and system-level simulation. pages 243–270, 2011.

[15] M. Taranetz, T. Blazek, T. Kropfreiter, M. K. Müller, S. Schwarz, and M. Rupp. Runtime precoding: Enabling multipoint transmission in lte-advanced system-level simulations. *IEEE Access*, 3:725–736, 2015.

[16] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. *Forecast and Methodology*, page 22, 2015.

[17] Naga Bhushan, Junyi Li, Durga Malladi, Rob Gilmore, Dean Brenner, Aleksandar Damnjanovic, Ravi Teja Sukhavasi, Chirag Patel, and Stefan Geirhofer. Network densification: The dominant theme for wireless evolution into 5G. *IEEE Communications Magazine*, 52(2):82–89, 2014.

[18] Mugen Peng. System Architecture and Key Technologies for 5G Heterogeneous Cloud\n Radio Access Networks. *IEEE Network*, 29(April):6–14, 2014.

[19] For Evaluation. Next Generation 5G Wireless Networks : A Comprehensive Survey. *Ieee Communications Surveys & Tutorials*, 18(3):2005–2008, 2016.

[20] Nenad Mladenovik, Jack Brimberg, Pierre Hansen, and Jos?? A. Moreno-P??rez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

[21] Nokia Siemens Networks. Mobile broadband with HSPA and LTE – capacity and cost aspects. page 12, 2010.

[22] Analysys Mason and Analysys Mason. 4 . 5 Should operators deploy their own backhaul , and what are the options ? pages 0–1, 2011.

[23] Mobile Backhaul. Backhauling with fibre ? (6), 2015.

[24] F Rossi, P Van Beek, and T Walsh. Handbook of Constraint Programming (Foundations of Artificial Intelligence). pages 281–322, 2006.

[25] Federico Barber and Miguel A Salido. Introducción a la programación de restricciones. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 7(20):13–30, 2003.

[26] Christian Schulte, Guido Tack, and Mikael Z. Lagerkvist. Modeling and Programming with Gecode. page 530, 2013.