



**Universidad**  
Zaragoza

## Trabajo Fin de Máster

Propuesta y evaluación de un sistema centralizado de  
selección de canal y asociación de usuarios en una red  
IEEE 802.11

Autor

Alejandro Latorre Alberich

Director

Jorge Ortín Gracia

Ponente

José Ramón Gállego Martínez

Escuela de Ingeniería y Arquitectura  
Máster en Ingeniería de Telecomunicación

Diciembre 2017





## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Alejandro Latorre Alberich

con nº de DNI 73000113E en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Máster \_\_\_\_\_, (Título del Trabajo)

Propuesta y evaluación de un sistema centralizado de selección de canal y  
asociación de usuarios en una red IEEE 802.11

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada  
debidamente.

Zaragoza, 24 de Noviembre de 2017

Fdo: \_\_\_\_\_



# **Propuesta y evaluación de un sistema centralizado de selección de canal y asociación de usuarios en una red IEEE 802.11.**

## **Resumen**

Este Trabajo Fin de Máster propone un algoritmo para realizar la asignación de canales a los puntos de acceso de una red IEEE 802.11 con tráfico TCP y la posterior asociación de los usuarios a los puntos de acceso para mejorar el funcionamiento global de la red.

Habitualmente, a cada punto de acceso se le asigna un canal en el momento en el que se instala, permaneciendo este canal invariable hasta que se vuelve a reconfigurar. Por otro lado, los usuarios se asocian al punto de acceso que captan con más señal. En este Trabajo Fin de Máster se propone un algoritmo centralizado que, en función del número y localización de los usuarios y del número y localización de los puntos de acceso, selecciona tanto el canal en el que opera cada punto de acceso como el punto de acceso al que se asocia cada usuario, siendo así capaz de optimizar el funcionamiento de la red.

El algoritmo propuesto se ha programado en C++, junto con una serie de funciones que nos facilitan la generación del escenario.

Dicho algoritmo se pone a prueba en diferentes escenarios programados en ns-3, el cual es un simulador de red de código libre ampliamente utilizado en el ámbito académico, donde se compara su rendimiento en diferentes situaciones con respecto a una asignación fija.

# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación y objetivos	3
1.2	Materiales y herramientas utilizadas	3
1.3	Organización de la memoria	4
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Acceso al medio en IEEE 802.11	5
2.2	Bandas frecuenciales en IEEE 802.11	8
2.3	Proceso de asociación de un STA a un AP	11
2.4	Estado del arte de mecanismos de selección de canal frecuencial y selección de AP	13
2.5	Propuesta	14
<b>3</b>	<b>Algoritmo de selección de canal frecuencial y asociación de usuarios</b>	<b>15</b>
3.1	Modelo de sistema	16
3.1.1	Estimación del <i>throughput</i> por flujo	18
3.2	Descripción del algoritmo	22
3.2.1	Teoría de juegos	22
3.2.2	Juego potencial	24

3.2.3 Solución del juego potencial	26
<b>4 Implementación</b>	<b>30</b>
4.1 Simulador ns-3	31
4.2 Generación de usuarios	32
4.3 Cálculo de matriz de cobertura	33
4.4 Cálculo de matriz de interferencia	35
4.5 Cálculo AP más cercano a cada usuario	36
4.6 Ejecución del juego	37
<b>5 Evaluación del algoritmo</b>	<b>38</b>
5.1 Parámetros ns-3	39
5.2 Evaluación del método de estimación del <i>throughput</i>	40
5.3 Evaluación del algoritmo de selección de canal frecuencial y asociación de usuarios	45
5.3.1 Índice de Jain	47
5.3.2 Suma total del <i>throughput</i>	50
5.3.3 Función de distribución acumulada	52
<b>6 Conclusiones y líneas futuras de trabajo</b>	<b>60</b>
6.1 Conclusiones	60
6.2 Líneas futuras	61
<b>Bibliografía</b>	<b>62</b>
<b>A Simulador de red n-s3</b>	<b>66</b>
A.1 Introducción a ns-3	66
A.2 Módulos de ns-3 utilizados	68

# Índice de Figuras

2.1	Reparto frecuencial de los diferentes canales en la banda de 2.4GHz	9
2.2	Espectro de los canales sin solape tanto con modulación DSSS como OFDM	10
2.3	Probe request	12
2.4	Probe response	12
3.1	Escenario número 1 para explicar el cálculo teórico de tasa efectiva	19
3.2	Escenario número 2 para explicar el cálculo teórico de tasa efectiva	20
3.3	Diagrama de bloques del juego implementado	27
5.1	Escenario número 1 en el que probar el cálculo teórico	40
5.2	Escenario número 2 en el que probar el cálculo teórico	42
5.3	Distribución hexagonal de los puntos de acceso	45
5.4	Asignación ideal de canal para evitar interferencias en una distribución hexagonal	46
5.5	Índice de Jain en función del número de con el triple de flujos ascendentes que descendentes	49
5.6	Índice de Jain en función del número de usuarios con el mismo número de flujos ascendentes que descendentes	49
5.7	Índice de Jain en función del número de con el triple de flujos descendentes que ascendentes	50
5.8	Throughput total en función del número de usuarios con el triple flujos ascendentes que descendentes	51
5.9	Throughput total en función del número de usuarios con el	



mismo número de flujos ascendentes que descendentes	51
5.10 Throughput total en función del número de usuarios con el triple de flujos descendentes que ascendentes	52
5.11 C.D.F con 5 flujos ascendentes y 15 descendentes	53
5.12 C.D.F con 10 flujos ascendentes y 10 descendentes	53
5.13 C.D.F con 15 flujos ascendentes y 5 descendentes	54
5.14 C.D.F con 10 flujos ascendentes y 30 descendentes	54
5.15 C.D.F con 20 flujos ascendentes y 20 descendentes	55
5.16 C.D.F con 30 flujos ascendentes y 10 descendentes	55
5.17 C.D.F con 15 flujos ascendentes y 45 descendentes	56
5.18 C.D.F con 30 flujos ascendentes y 30 descendentes	56
5.19 C.D.F con 45 flujos ascendentes y 15 descendentes	57
5.20 C.D.F con 20 flujos ascendentes y 60 descendentes	57
5.21 C.D.F con 40 flujos ascendentes y 40 descendentes	58
5.22 C.D.F con 60 flujos ascendentes y 20 descendentes	58
A.1 Módulos existentes en ns-3	68
A.2 Arquitectura de un Wi-FiNetDevice	69

## Índice de Tablas

4.1	Valores utilizados en ns-3	34
5.1	Parámetros Wi-Fi	39
5.2	Parámetros de flujo de transmisión	39
5.3	Throughput simulado vs Throughput calculado teóricamente en el escenario 1	41
5.4	Throughput simulado por flujo en el escenario 2	43
5.5	Throughput simulado agregado por AP vs throughput medio esperado por AP en el escenario 2	44

# Capítulo 1

## Introducción

Actualmente nos encontramos ante un entorno totalmente digitalizado. El sector de las telecomunicaciones crece permanentemente con el paso de los años y cada vez más personas requieren de las redes inalámbricas en su día a día, bien sea al utilizar su *smartphone* o su ordenador portátil. Debido a este crecimiento de dispositivos, se crean cada vez más y más redes inalámbricas.

Todos estos dispositivos cada vez son capaces de realizar más tareas y, por tanto, consumir más recursos de nuestra red (videojuegos en línea, *streaming*, subida de ficheros a la red etc.). Por tanto, necesitan que la red les ofrezca unas garantías de funcionamiento de los servicios para que se sientan satisfechos.

Debido a este auge en los dispositivos inalámbricos, cada vez se ofrece mayor conectividad Wi-Fi en lugares públicos tales como estaciones de tren, aeropuertos, bibliotecas etc. En estos escenarios, se suele disponer de varios puntos de acceso (*Access Points* – AP) colocados en diferentes puntos de la zona a cubrir.

Conforme aumenta el número de usuarios en una zona, el ancho de banda que le corresponde a cada usuario es menor, por lo que es necesario diseñar algoritmos que repartan el ancho de banda de un modo eficiente y equitativo. Por defecto, el usuario se asocia con el AP que recibe con mayor potencia (normalmente el más cercano) por lo que, si muchas

personas se ubican cerca de un AP, todos se asociarán con el mismo, aunque tengan cobertura de otros AP.

Por otro lado, en la banda de 2,4 GHz cada AP puede emitir en 14 canales diferentes (aunque en Europa solamente es posible transmitir en 13). Debido a la separación en frecuencia que hay entre estos canales, para que no exista interferencia, debe haber como mínimo una separación de 5 canales. Esto provoca que únicamente haya 3 canales que puedan ser empleados simultáneamente sin interferencia (por ejemplo, el 1, 6 y 11).

Este trabajo propone un algoritmo centralizado encargado de elegir el mejor canal para cada AP y asignar cada usuario al AP que mejor rendimiento le pueda aportar.

La validez de la solución propuesta se prueba mediante simulaciones, cuyos resultados muestran que nuestro algoritmo mejora los resultados obtenidos con otras soluciones.

## 1.1 Motivación y objetivos

La realización de este Trabajo Fin de Máster (TFM) se enmarca dentro del proyecto europeo WI-5 [1], englobado dentro del programa de investigación “Horizon 2020” de la Unión Europea [2]. El objetivo de este proyecto europeo es definir los casos de uso, los requerimientos y la arquitectura del sistema para conseguir una solución Wi-Fi que garantice la calidad de la red.

En este TFM, se propone una solución en la cual la asignación de canal para cada AP y la asociación de cada usuario a un AP se eligen de manera centralizada. Se busca que, en base a la ubicación de los AP y de los usuarios, el reparto del ancho de banda sea lo más eficiente y justo posible. Las pruebas se realizan con el simulador ns-3, basado en C++ y ampliamente utilizado por la comunidad científica.

Para comprobar si nuestra propuesta mejora la configuración por defecto, se utilizan varios escenarios en los que varían el número de usuarios, su ubicación y la cantidad de tráfico generado.

## 1.2 Materiales y herramientas utilizadas

Las herramientas empleadas en este Trabajo Fin de Máster son las siguientes:

- **ns-3:** Simulador de red de código libre. Permite simular redes Wi-Fi con varios AP y estaciones (STA) que envían tráfico de diversos tipos (UDP, TCP). Está escrito en C++. En el Anexo A se encuentra una descripción detallada de su estructura, así como de sus características principales.

- **Xcode:** Herramienta de desarrollo utilizado para programar el algoritmo que define la asignación de canales y de usuarios.
- **Microsoft Excel:** A la hora de realizar el procesado de los flujos se ha recurrido a Excel ya que nos proporciona la capacidad de realizar este procesado de los datos de manera sencilla y rápida.

### 1.3 Organización de la memoria

El contenido de la memoria se estructura de la siguiente forma:

- En el **Capítulo 1** se ha presentado el TFM y se ha dado una breve explicación de los objetivos del mismo.
- En el **Capítulo 2** se comenta el estado del arte y se explica el proceso de transmisión y asociación de los usuarios en una red Wi-Fi.
- En el **Capítulo 3** se propone el algoritmo centralizado y se explica su funcionamiento.
- En el **Capítulo 4** se explica cómo se ha realizado la implementación de la solución.
- En el **Capítulo 5** se explican los resultados obtenidos a través de simulaciones, comprobando el funcionamiento del algoritmo propuesto.
- En el **Capítulo 6** aparecen las conclusiones a las que se ha llegado tras realizar este trabajo y posibles líneas futuras a seguir.

# Capítulo 2

## Estado del arte

En este Capítulo se va a analizar el estado del arte respecto a la asignación de canales y la asociación de usuarios a cada AP.

En primer lugar, se va a explicar cómo se realiza el acceso al medio en una red Wi-Fi, lo que permitirá entender cómo se reparten los recursos entre los distintos usuarios de dicha red. También se analizarán los canales disponibles que tiene una red Wi-Fi. Posteriormente se describirán las principales soluciones en la bibliografía en lo que respecta a la asignación de canales a los AP de una red Wi-Fi y la asociación de usuarios a AP. Finalmente, se describirá someramente la solución propuesta para realizar ambos procesos simultáneamente (asignación de canal y asociación de usuarios a AP).

## 2.1 Acceso al medio en IEEE 802.11

Las redes basadas en IEEE 802.11 [3], comúnmente conocidas como redes Wi-Fi, son redes inalámbricas en las que el medio de transmisión se comparte entre todos los usuarios de la red. Esto hace necesario incorporar un mecanismo de control de acceso al medio que regule la transmisión de los distintos dispositivos conectados a la red.

Hay dos modos de funcionamiento en los que puede trabajar una red 802.11: Infraestructura y Ad-Hoc.

El modo infraestructura es un modo de funcionamiento que permite conectar equipos de usuario por medio de uno o varios AP, de modo que todas las comunicaciones pasan a través de ellos. La implementación de este tipo de red requiere poner AP diseminados en la zona que se quiere cubrir. Los AP y los equipos de usuario deben estar en el mismo canal y estar configurados con el mismo SSID (nombre de la red) para que puedan comunicarse.

El modo Ad-Hoc es un modo de funcionamiento que permite la comunicación directa entre equipos de usuario, sin necesidad de utilizar un AP. Este modo es ideal para poder interconectar rápidamente equipos entre ellos directamente (por ejemplo, para el intercambio de archivos entre PC portátiles en un tren). La implementación de una red de este tipo se limita a configurar los equipos en modo Ad-Hoc (en lugar del modo Infraestructura), la selección del canal (frecuencia) y de un SSID (nombre de la red) común a todos.

Este trabajo se centra en las redes que funcionan en tipo infraestructura ya que son las que se emplean más comúnmente y las que proporcionan acceso a Internet mediante Wi-Fi.



Los principales mecanismos para solucionar el problema del acceso al medio propuestos en Wi-Fi son CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) y RTS/CTS (Ready to Send / Clear to Send).

En CSMA/CA, cuando una estación (ya sea el AP o un equipo de usuario) quiere transmitir una trama, mira si el canal está libre, y en caso afirmativo, la transmite. Si el canal está ocupado porque hay otra estación transmitiendo, lanza una variable aleatoria que indica el número de slots (unidades de tiempo definidas en el estándar) que tiene que esperar antes de volver a intentar a transmitir. Este contador solo se reduce cuando se detecta que el canal está libre. Una vez que el contador llega a cero, la estación transmite. Una vez finalizada la transmisión, espera la recepción de un reconocimiento positivo (ACK por parte del receptor). Si este llega, la transmisión ha sido correcta. Si no, intenta volver a transmitir la trama repitiendo el proceso anterior, pero aumentando el valor máximo que puede tomar la variable aleatoria.

Para disminuir las colisiones (error que ocurre cuando dos estaciones transmiten a la vez) y mejorar la eficiencia energética, es posible emplear el mecanismo RTS/CTS. Para ello, cuando una estación ha detectado mediante el procedimiento CSMA/CA descrito en el párrafo anterior que puede transmitir, envía en primer lugar una trama RTS indicando el tiempo que va a ocupar el canal. El receptor contesta al RTS con una trama CTS indicando a la estación que puede mandar la trama de información. Seguidamente, dicha estación la manda. Este procedimiento sirve fundamentalmente para dos cosas. En primer lugar, se evita que el resto de estaciones estén comprobando continuamente si el canal queda libre, con el consiguiente ahorro energético. En segundo lugar, disminuye el efecto de las colisiones, ya que la longitud del RTS es menor que el de una trama normal, por lo que si se produjera una colisión se detectaría más rápido.

Se puede demostrar que, cuando las estaciones implementan los mecanismos descritos, se evita que ninguna monopolice el medio de

transmisión, repartiéndose el tiempo de transmisión de modo equitativo entre todas las estaciones.

Por otro lado, hay que tener en cuenta que el tiempo de transmisión que le toque a cada estación luego se va a tener que repartir entre todas los flujos en las que esa estación mande información, y que, como se ha dicho, en el reparto del tiempo de transmisión al AP le toca en promedio los mismos recursos que al resto de equipos de usuario. Este hecho puede provocar limitaciones en el caso de que en la red haya tanto tráfico ascendente (de los usuarios al AP) como descendente (del AP a los usuarios).

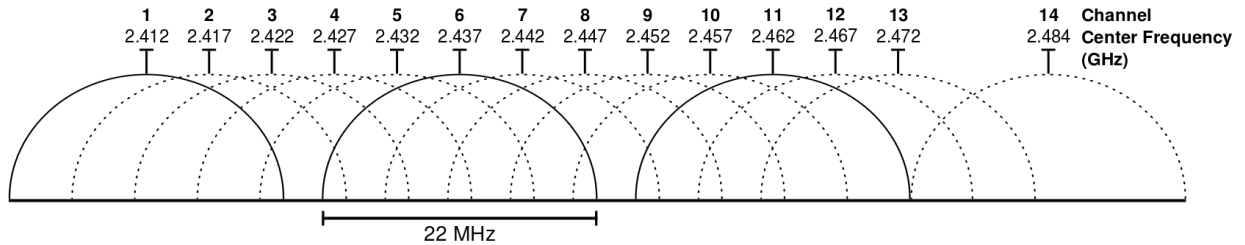
Por ejemplo, en un escenario con tráfico elástico, como TCP, que se adapta a los recursos disponibles, si se tuviera una red con dos estaciones, la primera de ellas con dos flujos ascendentes hacia el AP y la segunda con un flujo ascendente; y tres flujos descendentes desde el AP a las estaciones, el reparto del tiempo de transmisión sería de un tercio para la estación 1, un tercio para la estación 2 y un tercio para el AP. Esto implicaría que la estación 1 tendría que repartir ese tercio de su tiempo de transmisión entre sus dos flujos (un sexto del tiempo de transmisión para cada uno) y el AP tendría que repartir su tercio del tiempo de transmisión entre sus 3 flujos (un noveno para cada uno).

## **2.2 Bandas frecuenciales en IEEE 802.11**

La transmisión en las redes Wi-Fi se realiza de manera inalámbrica a través de ondas electromagnéticas. Dentro de las distintas versiones de Wi-Fi, los estándares 802.11b, 802.11g, y 802.11n-2.4 utilizan la banda de 2,400–2,500 GHz. Los estándares 802.11a y 802.11n pueden utilizar además la banda de 4,915–5,825 GHz. Comúnmente se conocen como bandas de 2,4GHz y 5GHz. El trabajo fin de máster se centra en la banda

de 2,4GHz, por ser la más limitante en cuanto a interferencia, si bien el algoritmo propuesto también sería aplicable a la banda de 5GHz.

La banda de 2,4GHz se divide en 14 canales, separados cada uno 5MHz, empezando en el canal número 1 cuya frecuencia central se encuentra en 2,412GHz. Se puede observar este reparto en la figura 2.1.



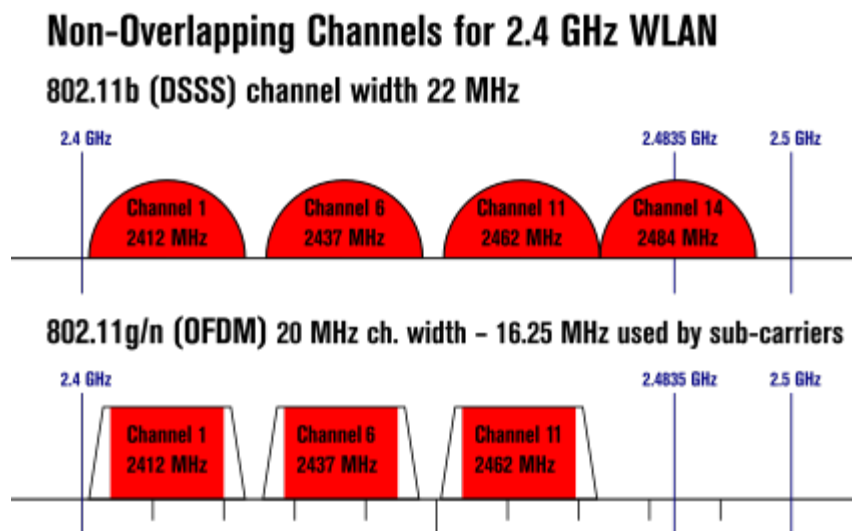
*Figura 2.1: Reparto frecuencial de los diferentes canales en la banda de 2.4GHz.*

Además de especificar la frecuencia central, 802.11 también especifica la máscara espectral, definiendo la distribución de potencia permitida a través de cada canal. La máscara requiere que la señal se atenúe un mínimo de 20dB de su pico de amplitud a una distancia de  $\pm 11$  MHz desde la frecuencia central. El ancho de banda de cada canal es por tanto es de 22 MHz. En consecuencia, para que dos canales no se solapen y, por tanto, no haya interferencias, se deben encontrar a una distancia de 5 canales.

La disponibilidad de los canales está regulada en cada país, esta restricción viene dada dependiendo de cómo utiliza cada país el espectro frecuencial. En un extremo, Japón permite el uso de los 14 canales para 802.11b y 13 canales para 802.11g/n. Otros países como España, al principio, solamente permitían el uso de los canales 10 y 11, aunque actualmente ya permiten el uso de 13 canales.

802.11b utiliza una modulación DSSS (Direct Sequence Spread Spectrum) y un ancho de banda de 22 MHz como se ha comentado anteriormente, resultando en 3 canales sin solape (1, 6 y 11).

802.11g utiliza una modulación OFDM (Orthogonal Frequency Division Multiplexing) y utiliza un ancho de banda de 20 MHz. OFDM es una técnica de transmisión que consiste en la multiplexación de un conjunto de ondas portadoras de diferentes frecuencias, donde cada una transporta información, la cual es modulada en QAM o en PSK. En la Figura 2.2 se puede observar los canales sin solape cuando se utiliza DSSS o OFDM.



*Figura 2.2: Espectro de los canales sin solape tanto con modulación DSSS como OFDM.*

## 2.3 Proceso de asociación de un usuario a un AP

En este apartado se va a explicar el proceso que sigue un usuario para asociarse con un AP, el cual consta de los siguientes pasos:

- 802.11 “probing”: En primer lugar, el usuario realiza la búsqueda de una red específica. Esto implica enviar una petición de prueba en múltiples canales que especifican el nombre de la red (SSID) y la tasa de bit.
- 802.11 “authentication”: Originalmente, las redes 802.11 fueron desarrolladas con dos mecanismos de autenticación. El primero, llamado “open authentication”, es básicamente una autenticación nula en la cual el usuario pide al AP que le autentique y el AP responde afirmativamente. Sería lo que se conoce como red abierta. El segundo tipo de autenticación, con seguridad WEP/WPA/WPA2, es un mecanismo de clave compartida que es ampliamente usado en redes pequeñas o en hogares.
- 802.11 “association”: Esta fase comienza cuando ya se ha superado la autenticación. En esta fase se acuerda la tasa de bit y establece el enlace de datos entre el usuario Wi-Fi y el AP. Si un usuario se había unido previamente a la red y se mueve de un AP a otro dentro de la misma red, la asociación se llama reasociación. La principal diferencia entre ambas es que, en el caso de la reasociación, el usuario envía la dirección MAC (BSSID) del AP anterior en su petición de reasociación para que el nuevo AP conozca la información previa del nuevo usuario.

En las redes Wi-Fi convencionales, los AP informan sobre su presencia mediante el envío de unas tramas llamadas *beacon* que incluyen el SSID y el BSSID. Antes de la asociación, los usuarios recogen información sobre los AP mediante el escaneo de los canales uno a uno o bien mediante escaneo pasivo o activo. En modo pasivo, el usuario se mueve a lo largo de

todos los canales y espera en cada uno a escuchar los *beacon* de cada canal. El usuario busca *beacon* que contengan un SSID al que se haya podido conectar previamente. Si recibe *beacon* de múltiples AP con el mismo SSID, intenta conectarse al AP con la mayor potencia recibida (normalmente al AP más cercano).

Los usuarios también realizan escaneo activo, en el cual el usuario envía tramas de *probe request* en cada canal. Estas tramas pueden contener un SSID de una red específica que esté buscando o también puede buscar cualquier SSID que se encuentre próximo al usuario. Los AP responden a estas tramas con otras tramas llamadas *probe response*. El contenido de éstas es similar a las tramas *beacon*. Los AP que se encuentran en un canal determinado responden a la *probe request* con una *probe response* con el SSID, las tasas soportadas y la seguridad que incluye.

Se puede ver en las Figuras 2.4 y 2.5 un ejemplo en el que un usuario está buscando activamente el SSID “wpa1” y su consiguiente *probe response*.

```
IEEE 802.11 wireless LAN management frame
  Tagged parameters (31 bytes)
    SSID parameter set: "wpa1"
    ...
    Supported Rates: 1.0(B) 2.0(B) 5.5 11.0 6.0 9.0 12.0 18.0
    ...
    Extended Supported Rates: 24.0 36.0 48.0 54.0
```

*Figura 2.3: Probe request.*

```
...
IEEE 802.11 wireless LAN management frame
...
  Tag Number: 1 (Supported Rates)
  Tag length: 8
  Tag interpretation: Supported rates: 1.0 2.0 5.5 11.0(B) 6.0 9.0 12.0 18.0
[Mbit/sec]
  ...
  Tag interpretation: WPA IE, type 1, version 1
  Tag interpretation: Multicast cipher suite: TKIP
  Tag interpretation: # of unicast cipher suites: 1
  Tag interpretation: Unicast cipher suite 1: TKIP
  Tag interpretation: # of auth key management suites: 1
  Tag interpretation: auth key management suite 1: WPA
  Tag interpretation: Not interpreted
...
```

*Figura 2.4: Probe response.*

## 2.4 Estado del arte de mecanismos de selección de canal frecuencial y de selección de AP

El problema que aparece respecto a la asignación de canal es que solamente hay 3 canales disponibles sin interferencia para poder transmitir (canales 1, 6 y 11) por lo que es muy importante definir de manera correcta en qué canal debe transmitir cada uno de los AP para poder minimizar la interferencia y mejorar la eficiencia de la red. Por este motivo, la interferencia entre los diferentes AP en una red Wi-Fi es un reto muy importante que se ha investigado de manera extensa en la literatura. Por ejemplo, en [5], los autores proponen una solución donde un AP puede elegir el canal en función de los AP vecinos que detecta y en la evaluación local de los canales escaneados menos congestionados. En [6], los autores proponen un algoritmo heurístico que asigna un canal a un AP analizando el efecto del solape parcial de los canales en el throughput de la red Wi-Fi. En [7], los autores formulan la optimización de la asignación de canales como un “graph colouring problem” [8].

Por otro lado, el proceso estándar que siguen los clientes para asociarse con los AP se basa únicamente en la potencia recibida y no tiene en cuenta el estado de la red a la hora de tomar la decisión de a qué AP asociarse, lo cual puede hacer que haya AP muy congestionados y otros cercanos con poco tráfico. Estas limitaciones también han hecho que el problema de selección de AP se haya estudiado de manera extensa en la literatura. De manera genérica, las distintas propuestas existentes se pueden clasificar en distribuidas [9] o centralizadas [10]. En un sistema distribuido, un cliente recopila medidas de rendimiento de la red antes de elegir el AP más acorde teniendo en cuenta una métrica específica. Por otro lado, los sistemas centralizados confían en una visión global obtenida del controlador de red para decidir cuál es el AP más adecuado. A pesar de que requieran mayor señalización, estas soluciones centralizadas tienden a ser más eficientes, especialmente en redes Wi-Fi grandes, ya que el controlador central es capaz ya no solo de obtener una visión más precisa del estado de la red completa, sino también de aplicar un balanceo de carga para evitar la

congestión de ciertos AP. Esta solución también se pueden clasificar dependiendo de sus propósitos. Así, hay algoritmos que se basan en una maximización de una métrica de rendimiento como puede ser el throughput [9], la eficiencia del ancho de banda [11] o la mínima atenuación debido al path-loss [12]. Aparecen también modelos que se basan en múltiples métricas para decidir cuál es la mejor elección [13] y finalmente aparecen también otras soluciones que están basadas en balancear la carga entre los diferentes AP que componen una red Wi-Fi [10].

## 2.5 Propuesta

La propuesta de este trabajo consiste en el diseño de un sistema centralizado capaz de realizar conjuntamente las dos tareas descritas anteriormente: la selección de canal de cada AP y la asignación de usuarios a AP. El algoritmo se diseña para actuar de manera dinámica, de modo que conforme se vayan modificando las condiciones de la red (los usuarios se mueven, aparecen nuevos flujos etc.), se efectúe tanto la reasignación de usuarios como la de canales si fuera necesario.



# Capítulo 3

## Algoritmo de selección de canal frecuencial y asociación de usuarios

En este capítulo se va a describir el algoritmo diseñado en este TFM.

En primer lugar, se explica el modelo de red sobre el que se va a evaluar el algoritmo. A continuación, se describe el algoritmo propiamente dicho, el cual se basa en la Teoría de Juegos, y más concretamente, en los juegos potenciales. Finalmente se explicará cómo ha sido implementado este algoritmo en C++ junto con el resto de programas que permiten crear el entorno de simulación para probar el algoritmo en ns-3.

### 3.1 Modelo de sistema

La popularidad de la tecnología Wi-Fi ha hecho que aparezca en lugares y espacios públicos como aeropuertos, estaciones de tren y campus de universidades. Estas redes son creadas mediante el despliegue de AP solapados geográfica y frecuencialmente para garantizar una buena cobertura y dar conectividad a los usuarios. A diferencia de las redes cableadas donde el tamaño de una red está fijado y la cantidad de tráfico puede ser predicho, las redes Wi-Fi son muy dinámicas ya que los usuarios pueden unirse y marcharse en cualquier momento.

El escenario que se va a estudiar en este trabajo consiste en a una zona geográfica en la que hay desplegados una serie de AP para dar cobertura Wi-Fi, los cuales están controlados por un administrador común y en la que hay también una serie de usuarios desplegados de manera aleatoria a lo largo de la zona geográfica que se quieren conectar a los diferentes AP.

Para poder utilizar un controlador central, éste debe conocer el estado de la red. En concreto, en el algoritmo propuesto, tendrá que calcular una matriz de interferencias entre los AP para saber para cada AP el resto de AP que le interfieren o que interfiere. Además, también tendrá que calcular una matriz de cobertura que indique qué AP puede elegir cada uno de los usuarios.

Para poder implementar estos dos cálculos en un sistema real, se requiere que cada AP disponga de dos transceptores radio en la misma banda frecuencial (la de 2,4 GHz en el caso de este trabajo). De este modo, uno de ellos se utiliza para realizar la transmisión propiamente dicha, mientras que el otro se emplea para monitorizar todas las bandas frecuenciales y realizar las medidas necesarias.

Concretamente, para calcular la matriz de interferencias, se debe ver qué *beacon* es capaz de decodificar cada AP y, por tanto, saber con qué AP está a una distancia en la que se interferiría en caso de trabajar en el mismo canal. Por otro lado, para el cálculo de la matriz de cobertura, cada

AP debe monitorizar las transmisiones de los usuarios en las distintas bandas frecuenciales para determinar a qué podría dar servicio (asumiendo un canal de propagación simétrico, si se recibe la transmisión de un usuario, también se podría transmitirle). Recopilando esta información para cada AP se puede calcular el conjunto de AP que puede servir a cada usuario.

En este escenario genérico puede existir tráfico tanto ascendente, desde los usuarios al AP, como descendente, desde el AP a los usuarios. Dicho tráfico puede deberse a aplicaciones muy diversas, que pueden funcionar tanto sobre TCP como UDP. Dado que el mayor porcentaje de tráfico existente en Internet funciona sobre TCP, como primera aproximación este trabajo se ha centrado únicamente en este tipo de flujos. No obstante, la inclusión de tráfico UDP queda pendiente como una futura línea de trabajo.

Para valorar si se están aprovechando los recursos de la red, se pueden utilizar distintas métricas. Como primera aproximación, se puede intentar maximizar el throughput total entregado por la red. Este tipo de métrica puede provocar que los recursos no se repartan de una manera equitativa entre los distintos usuarios. Por otro lado, se puede intentar maximizar otras funciones que sean más justas a la hora de repartir el canal, como podría ser maximizar la suma del logaritmo del throughput de cada uno de los flujos, lo cual se conoce comúnmente como *proportional fairness* [14].

En cualquier caso, y dado que los flujos TCP se ajustan al ancho de banda disponible en cada momento, para poder desarrollar un algoritmo que permita maximizar estas funciones, se debe poder estimar de alguna manera las tasas que le corresponden a cada uno de los flujos TCP en función del estado de la red (en qué canal transmite cada AP y a qué AP están asociados los usuarios). A partir de esas estimaciones se puede intentar maximizar las funciones de una manera teórica y luego proceder a la configuración en el escenario real. Para ello en el siguiente apartado se explica cómo se realiza esa estimación.

### 3.1.1 Estimación del throughput por flujo

Como se ha comentado anteriormente, el TFM se considera únicamente la presencia de tráfico TCP. Debido a la naturaleza del protocolo TCP, el ancho de banda que tiene cada usuario se divide por igual entre todas sus conexiones TCP.

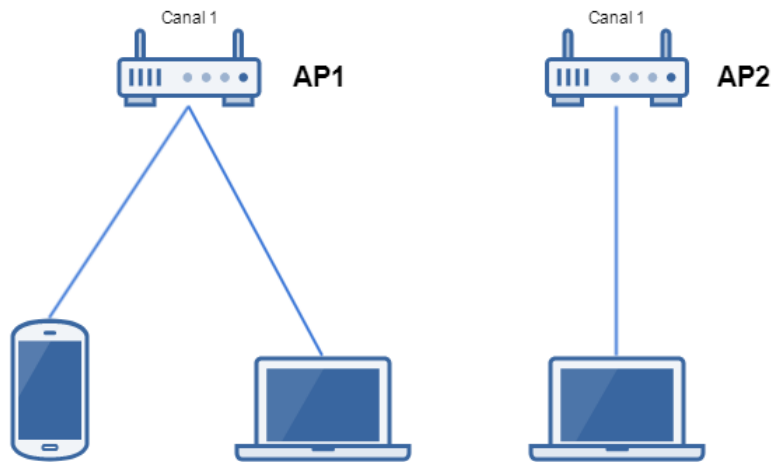
Asumido este comportamiento y con el reparto del tiempo de transmisión entre los usuarios de un AP que implica CSMA/CA (reparto equitativo entre usuarios), se puede hacer una aproximación del tiempo de transmisión (y por tanto del ancho de banda) que le corresponderá a cada flujo TCP en función del número de conexiones que tiene un usuario, del número de usuarios que hay en cada AP, y de las interferencias entre AP.

Como también se ha comentado anteriormente, en el escenario considerado hay tanto tráfico ascendente como descendente. Se asume también que se dispone de la matriz de interferencia entre los AP, el canal en el que transmite cada AP y los usuarios que tiene asociados. El objetivo es, dada la topología y conociendo los flujos TCP ascendentes y descendentes que tiene cada usuario, determinar cuál es el ancho de banda que le correspondería a cada una.

El recurso de transmisión del que dispone cada AP es el tiempo, el cual se reparte entre el número de usuarios que tenga conectados. Es decir, si hay dos usuarios asociados a un AP, cada uno de ellos dispondrá del canal un total 50% del tiempo. Si cada usuario tuviera un único flujo TCP ascendente, a cada una de ella le correspondería el 50% del tiempo de transmisión. Si por ejemplo el segundo usuario tuviera dos flujos ascendentes, al flujo TCP del primer usuario le correspondería el 50% del tiempo y a cada una de los flujos del segundo usuario le correspondería el 25% del tiempo.

Por otro lado, en caso de que dos o más AP estén en el mismo canal y a una distancia a la que se interfieran, todos ellos compartirán el canal, por lo que el tiempo de transmisión se tiene que repartir entre todos ellos. Por

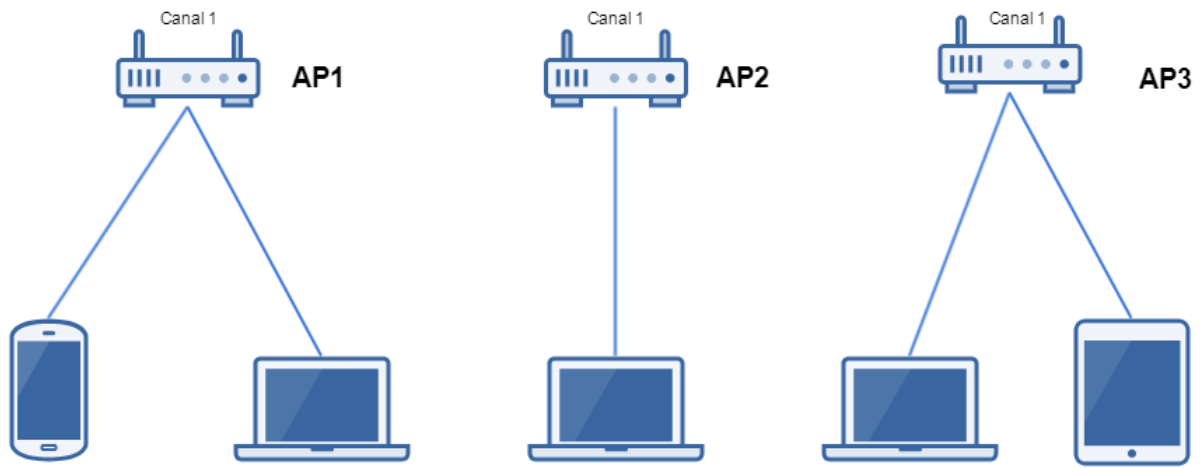
ejemplo, en el escenario de la Figura 3.1. hay dos AP que se interfieren entre sí. En el AP1 hay asociados un móvil y un ordenador y en el AP2 hay asociado un ordenador. Suponemos que cada uno de los 3 dispositivos tiene una única conexión TCP ascendente.



*Figura 3.1: Escenario número 1 para explicar el cálculo teórico de tasa efectiva.*

Como en total hay tres dispositivos transmitiendo, a cada uno de ellos le corresponde un tercio del tiempo de transmisión disponible, lo que implica que los usuarios conectados al AP1 están empleando dos tercios de los recursos de transmisión y el usuario conectado al primer AP consume un tercio de los recursos de transmisión.

Consideremos ahora un ejemplo un poco más complejo (Figura 3.2). En este caso tenemos 3 AP, el AP1 se interfiere con el AP2, el AP2 se interfiere con los AP1 y AP3 y el AP3 se interfiere con el AP2.



*Figura 3.2: Escenario número 2 para explicar el cálculo teórico de tasa efectiva.*

Con lo que se ha comentado, el tiempo de transmisión del AP1 se tiene que repartir entre 3 usuarios, el tiempo de transmisión del AP2 se tiene que repartir entre 5 usuarios y el tiempo de transmisión del AP 3 se tiene que repartir entre 3 usuarios. Por tanto, a los dos usuarios del AP1 les correspondería un tercio del tiempo de transmisión del AP1, al usuario del AP2 le correspondería un quinto del tiempo de transmisión del AP2 y a los dos usuarios del AP3 le correspondería un tercio del tiempo de transmisión del AP3.

No obstante, si el único usuario del AP2 solo puede transmitir un 20% del tiempo, el tiempo de transmisión que podrían transmitir los usuarios de los AP1 y AP3 es superior al estimado originalmente. Para obtener cuál es el tiempo de transmisión que obtendrían realmente, hay que fijar el tiempo de transmisión del único usuario del AP2 (al que le corresponde un tiempo de transmisión menor) y repartir el tiempo restante entre los usuarios del resto de AP. Por ejemplo, en el AP1, el usuario del AP2 consume un 20% del tiempo, por lo que el tiempo restante es del 80% y a cada uno de sus usuarios le correspondería un 40% del tiempo de transmisión. Lo mismo ocurriría en los usuarios del AP3.

Este tipo de reparto se conoce como “max-min fairness” [10] y se puede obtener un procedimiento cerrado para calcular cuál es el tiempo de transmisión que le correspondería a cada estación.

Para ello, en primer lugar, se calcula el número de equipos entre los que tiene que repartir su tiempo de transmisión cada AP. Este número es la suma del número de equipos que tiene asociado (contándose también el AP si tiene algún flujo descendente) más los equipos que tienen asociados los AP que le interfieren (contándose también los AP si tienen algún flujo descendente). Posteriormente se ordenan los AP por orden descendente de este valor y, siguiendo este orden, se aplica la siguiente expresión para obtener el tiempo de transmisión que le tocaría a cada uno de los equipos de cada AP:

$$T_i = \frac{1 - \sum_{j < i, j \in I_j} n_j T_j}{n_i + \sum_{j > i, j \in I_j} n_j}$$

donde  $T_i$  es el tiempo de transmisión que corresponde a cada uno de los equipos asociados al AP  $i$ ,  $n_i$  es el número de equipos que tiene asociado el AP  $i$  (contándose también el AP si tiene algún flujo descendente) e  $I_j$  es el conjunto de AP que interfieren al AP  $i$ . Cuando  $i = 1$ , el numerador de esta expresión es 1.

De esta manera, se puede calcular el tiempo de transmisión que le corresponde a cada estación. Dicho tiempo de transmisión se tiene que dividir entre el número de flujos que está transmitiendo dicha estación. Finalmente, la tasa de transmisión efectiva que tendrá cada flujo sería el producto del tiempo de transmisión que le corresponde por la tasa física a la que puede transmitir dicho flujo, la cual depende de las condiciones del canal de transmisión.

$$c_j = r_k \cdot T_i / m_k$$

donde  $c_j$  es el throughput que es capaz de transmitir el flujo  $j$ ,  $T_i$  es el tiempo de transmisión que le corresponde al equipo  $k$  que transmite el flujo

$j$  asociado al AP  $i$ ,  $m_k$  es el número de flujos que transmite dicho equipo y  $r_k$  es la tasa negociada entre el usuario  $k$  y el AP  $i$ .

## 3.2 Descripción del algoritmo

El objetivo del algoritmo que se describe a continuación es realizar una gestión eficiente de los recursos disponibles en la red por medio de asignar o re-asignar a los usuarios a los AP más adecuados y elegir en cada AP el mejor canal para reducir la interferencia con otros AP. Para ello, en primer lugar hay que estudiar el escenario analizado y así decidir qué tipo de algoritmo se puede implementar.

Idealmente, dada una métrica global a maximizar (*throughput* total, suma de los logaritmos de los *throughput*) podría plantearse buscar la solución óptima de manera dinámica cada vez que se produce cambio en la red. Sin embargo, esto no es viable computacionalmente hablando. Por lo tanto, nuestro objetivo es buscar un algoritmo heurístico subóptimo. Dentro de estos algoritmos se encuentran los juegos potenciales que, dadas sus propiedades, tal como se va a detallar a continuación, son unos buenos candidatos para conseguir dicho objetivo.

### 3.2.1 Teoría de juegos

Se va a modelar la selección de canal y la selección de AP por parte de los usuarios como un juego formal, desarrollando un algoritmo definiendo una serie de jugadores, de estrategias y de una función de utilidad [15].

De manera formal un juego  $\Gamma$  se define como  $\Gamma = \{P, \{S_i\}_{i \in P}, \{u_i\}_{i \in P}\}$ , donde  $P$  es el vector finito de jugadores,  $S_i$  es el conjunto de estrategias de cada jugador  $i$  y  $u_i: S \rightarrow R$  es la función de utilidad de cada jugador, con  $S = \times_{i \in P} S_i$  el espacio de estrategias del juego, formado por el producto cartesiano de la serie de estrategias de cada jugador en el juego.



La función de utilidad  $u_i$  es una función de  $s_i$ , la estrategia elegida por el jugador  $i$ , y de  $s_{-i}$ , la estrategia del resto de jugadores del juego. Cada jugador elegirá aquella estrategia que mejore su función de utilidad considerando las estrategias actuales del resto de los jugadores.

Una de las claves cuando se diseña un juego es la elección de  $u_i$  de tal manera que la acción de los jugadores provoque un rendimiento bueno en general. Además, en este caso es también deseable la existencia de un punto de equilibrio para asegurar la convergencia de los juegos propuestos cuando se realiza la optimización. En este contexto, es útil el concepto de equilibrio de Nash, definido como la situación donde ningún jugador puede mejorar sin que lo haga ninguno de los demás jugadores. Por tanto, un equilibrio de Nash de un juego  $\Gamma$  es un perfil  $s^* \in S$  de acciones tal que para cada jugador  $i \in P$  se tiene

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$$

Para todo  $s_i \in S$ , donde  $s_i$  denota cualquier estrategia de un jugador  $i$  diferente de  $s_i^*$  y  $s_{-i}^*$  denota las estrategias de todos los jugadores que no sean el jugador  $i$  en el perfil  $s^*$ .

Para plantear el juego que ayude a realizar la optimización, es necesario definir los jugadores, las posibles estrategias de cada uno de ellos y la función de utilidad.

En este caso, habrá de dos tipos de jugadores, los AP y los usuarios. Cada uno de ellos tiene unas estrategias posibles diferentes. La estrategia de los AP será elegir en qué canal transmitir (1, 6 u 11) y la de los usuarios será la de a qué AP asociarse. Para explicar la función de utilidad del juego, primero se va a explicar qué es un juego potencial en el siguiente apartado ya que la función de utilidad propuesta cumple los requisitos para que el juego se pueda considerar como potencial.

### 3.2.2 Juego potencial

Un juego potencial exacto [16] es un juego para el cual existe una función potencial  $V : S \rightarrow R$  tal que

$$\begin{aligned}\Delta u_i &= u_i(s_i, s_{-i}) - u_i(s'_i, s_{-i}) = \Delta V \\ &= V(s_i, s_{-i}) - V(s'_i, s_{-i}) \quad \forall i \in P, \forall s_i, s_{-i} \in S_i.\end{aligned}$$

Esta definición implica que cada interés individual de cada jugador está alineado con el interés del grupo (la función potencial) por lo que cada cambio en la función de utilidad de un jugador ( $u_i$ ) se ve directamente reflejado en el mismo cambio para la función potencial ( $V$ ). Así, cualquier jugador que elige una estrategia que mejora su utilidad individual dadas todas las estrategias actuales del resto de los jugadores llevará directamente a una mejora en el valor de la función potencial. Con esta definición, se puede demostrar que si solamente un jugador actúa en cada paso (un juego secuencial repetido) y cada jugador maximiza (“*best response strategy*”) o al menos mejora (“*better response strategy*”) su utilidad dadas las acciones del resto de los jugadores, entonces el proceso siempre converge en un número de pasos finitos a un equilibrio de Nash. En este tipo de juegos también se cumple que los máximos globales de la función potencial  $V$  son equilibrios de Nash, a pesar de que puede ser solamente un subconjunto de todos los equilibrios de Nash del juego, es decir, no todos los equilibrios de Nash son máximos globales de la función potencial.

Estas propiedades interesantes de los juegos potenciales (convergencia garantizada en un número de pasos finito y la posibilidad de maximizar la función potencial) hacen de ellos una herramienta útil para realizar una optimización distribuida. Una importante limitación de modelar nuestro escenario como un juego potencial es que los jugadores requieren tener la información total del estado de la red. Sin embargo, como se ha indicado previamente, la solución propuesta se basa en un sistema centralizado que se encargará de obtener toda la información de todos los jugadores y

realizar el juego también de forma centralizada. Es decir, a pesar de que formalmente los jugadores son los usuarios y los AP, realmente es el controlador central el que, con la información recopilada de todos ellos, ejecuta el juego.

A continuación, se muestra cómo se puede diseñar específicamente un juego para que esa potencial: Si se define la función de utilidad igual a la función potencial, se consigue que todos los jugadores tengan como función de utilidad la de mejorar la condición global de la red y por tanto todos tengan el mismo objetivo.

En este TFM, se van a contemplar dos funciones potenciales y, por tanto, dos funciones de utilidad: una para maximizar la tasa efectiva total de la red y otra para maximizar la suma de logaritmos de las tasas efectivas de cada uno de los flujos para que el reparto sea más equitativo. Así, las dos funciones potenciales que se van a utilizar son:

- Maximizar tasa efectiva total de la red:

$$u_i(s_i, s_{-i}) = \max \sum_{j=1}^N c_j$$

Donde  $c_j$  es igual a la tasa disponible en el flujo  $j$  y  $N$  es el número total de flujos en la red. Dicha tasa se estima tal como se ha descrito en el apartado 3.1.1. a partir del tiempo de transmisión estimado correspondiente al flujo  $j$  y la tasa física a la que puede transmitir dicho flujo.

De esta manera, se maximiza la tasa efectiva total que se consigue en la red sin tener en cuenta que algunos usuarios puedan conseguir mucha tasa mientras que otros se pueden quedar con nada o con muy poca.

- Maximizar suma de logaritmos

$$u_i(s_i, s_{-i}) = \max \sum_{j=1}^N \log(c_j)$$

Donde  $c_j$  es igual a la tasa efectiva disponible en el flujo  $j$ .

De esta manera, maximizando la suma de logaritmos, se penaliza las situaciones en las que hay muchos usuarios a los que no se les asigna casi tasa efectiva y, por tanto, se consigue que el reparto sea más justo entre todos.

Previamente ya se ha comentado cómo se estiman teóricamente los valores de  $c_j$  en función del número de flujos por AP y la interferencia entre AP.

### 3.2.3 Solución del juego potencial

En este trabajo, se ha implementado un juego secuencial repetido en el que en cada iteración del mismo solo un jugador intenta modificar su estrategia. Se ha implementado una estrategia *better-response*, es decir, en que el jugador encuentra una estrategia que mejora su utilidad, la elige y deja de buscar en el espacio de estrategias.

Como se ha comentado anteriormente, en este juego hay dos tipos de jugadores: los AP, cuya estrategia es la elección del canal, y los usuarios, cuya estrategia es la elección del AP al que conectarse. El juego se ha implementado de manera que en cada ciclo (una ronda en la que todos los jugadores intentan modificar su estrategia) en primer lugar juegan todos los AP pudiendo variar su canal y posteriormente juegan todos los usuarios (STA en la figura 3.3) pudiendo elegir a qué AP conectarse.

A continuación, en la figura 3.3 se muestra un diagrama que ilustra el funcionamiento del juego completo.

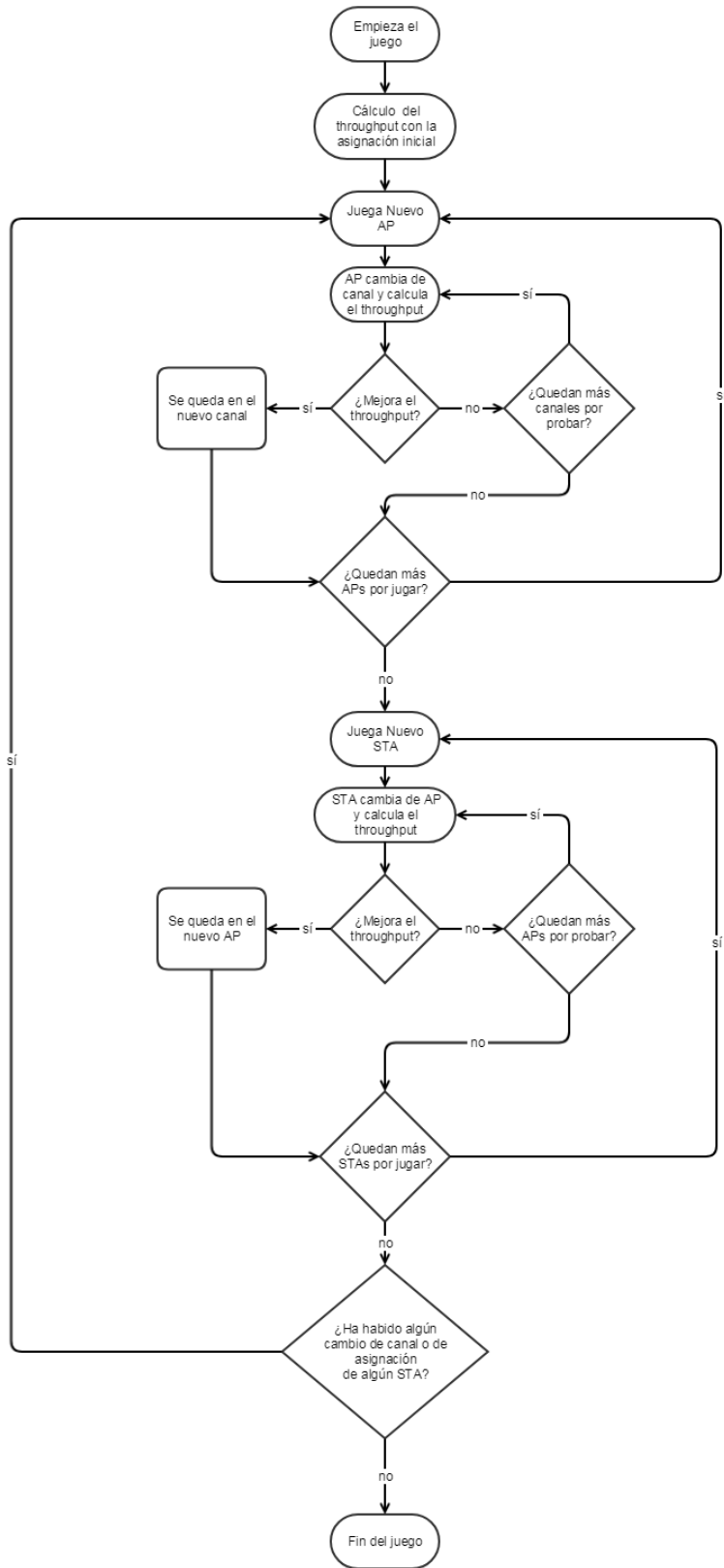


Figura 3.3 Diagrama de bloques del juego implementado.

El funcionamiento es el siguiente:

- 1) En primer lugar, se asume que inicialmente cada usuario se conecta a su AP más cercano y se calculan las tasas que se conseguirían con dicha asociación por defecto. Se considera que estas son las condiciones iniciales con las que se inicia el juego.
- 2) En el juego, comienzan jugando los AP. Cada uno de ellos varía el canal en el que se encuentra y recalcula su función de utilidad. Si dicha función de utilidad ha aumentado, cambia a ese canal y juega el siguiente AP. Si no ha aumentado, sigue probando hasta que no hay más canales por probar. Si no ha mejorado con ninguno, se queda en el que estaba y pasa a jugar el siguiente AP. Cuando ya no quedan más AP por jugar, comienzan a jugar los usuarios.
- 3) Basándose en la matriz de coberturas, el usuario va cambiando de AP al que asociarse y recalcula el valor de la función de utilidad correspondiente a dicha asociación. Como ocurre anteriormente en los AP, si el valor de la función de utilidad aumenta al cambiar de AP, se queda asociado a dicho AP. Si no mejora, se mantiene en el AP que estaba. Cuando ha mejorado o una vez que se han evaluado todos los posibles AP sin mejorar, pasa el turno al siguiente usuario. Así sucesivamente hasta que no quedan más usuarios por jugar.
- 4) Si tras jugar todos los AP y todos los usuarios, la función de utilidad de alguno de ellos ha aumentado con respecto a la configuración previa, se vuelve a repetir el proceso completo. Si, por el contrario, no se ha producido ningún cambio y por tanto ningún jugador ha mejorado, el proceso ha convergido a un equilibrio de Nash y se da por terminado el juego.

Dadas las propiedades descritas en el apartado 3.2.2, este procedimiento asegura que siempre se converge a un equilibrio de Nash. Dicho equilibrio no tiene por qué ser el óptimo, pero será al menos un máximo relativo.

Tras finalizar el algoritmo, se conoce el canal en el que se tiene que configurar cada AP y a qué AP se tiene que asociar cada uno de los usuarios dadas las condiciones de la red en cada momento.

En el capítulo siguiente se explica cómo se ha implementado este algoritmo y cómo se ha integrado dentro del simulador de red ns-3.

# Capítulo 4

## Implementación

En este capítulo se describe cómo se ha implementado el escenario completo en el que se ha evaluado el algoritmo propuesto. Primero se explica qué es ns-3 [4] y por qué se ha decidido utilizarlo. Posteriormente en los siguientes apartados se describen los programas que se han desarrollado para simular el escenario completo.



## 4.1 Simulador ns-3

El simulador ns-3 es un simulador de redes basado en eventos discretos, que se emplea principalmente para investigación y educación. El proyecto ns-3 se inició en el año 2006, y es un proyecto de desarrollo de código abierto bajo licencia GNU GPLv2.

El simulador ns-3 proporciona una plataforma de simulación implementada principalmente en C++, aunque algunas de sus estructuras están escritas en Python. Es compatible con Linux, Mac OS y FreeBSD.

Este simulador proporciona modelos para trabajar con paquetes de datos y redes, además de incorporar modelos de movilidad y propagación. Todo ello forma un motor de simulación en el que los usuarios pueden programar sus simulaciones, de forma que se pueden realizar estudios sobre el comportamiento de los sistemas cuando no se dispone de los dispositivos físicos.

Al estar organizado como una librería y estar escrito en C++ permite realizar enlaces estáticos y dinámicos. Una de sus características más importantes y que más se ha utilizado en este trabajo es el hecho de permitir la definición de nuevas topologías, escenarios y modelos de red, que pueden ser fácilmente integrados y depurados en el simulador.

En este trabajo, este simulador se ha utilizado para realizar la simulación de las diferentes topologías de red que se muestran en el capítulo de resultados. Cabe destacar que si bien permite simular redes Wi-Fi, no incluye todavía todas las funcionalidades del estándar, lo que ha supuesto ciertas limitaciones a la hora de realizar las pruebas.

Una parte importante del tiempo dedicado en el desarrollo de este trabajo se ha invertido en entender cómo funciona el simulador y al estudio de cómo programar las diferentes topologías y extraer la información de interés ya que las posibilidades del simulador son muy amplias y en

primera aproximación se hacía complicado encontrar los módulos que coincidían con los requisitos de este trabajo.

Además, se ha escrito un código lo más parametrizado posible para que simplemente variando los valores de entrada que definan la posición de los AP, de los usuarios, el canal en el que transmiten los AP y si el usuario transmite flujo ascendente o descendente, se pueda simular el escenario automáticamente. De esta manera se simplifica la realización de nuevas simulaciones sin que la programación en ns-3 suponga un retraso adicional.

A continuación, se explican el resto de funciones que se han desarrollado para evaluar el algoritmo propuesto. Tal como se ha descrito anteriormente, el objetivo del algoritmo es que se aplique de manera dinámica conforme cambian las condiciones de la red. Por lo tanto, idealmente, todas las funciones que se definen a continuación se deberían integrar en el propio código de ns-3 para que, conforme variase el escenario (entran nuevos usuarios en la red, se mueven los usuarios por el escenario etc.) se pueda recalcular la asignación de recursos. Por esta razón, todas ellas se han programado en C++. Sin embargo, en este trabajo y como primera aproximación, se ha optado por estudiar el algoritmo en un escenario estático, en el que todos los flujos de los usuarios están activos durante toda la simulación y no existe movilidad. Por este motivo, las funciones no están integradas dentro del simulador si no que proporcionan los parámetros de entrada del mismo, dejando como línea futura su integración en el simulador.

## 4.2 Generación de usuarios

En primer lugar, se deben generar usuarios de manera aleatoria. Para ello, se ha desarrollado un programa en el que se puede definir el área sobre la que desplegar los usuarios y el número de usuarios a generar. Tras ejecutarlo, se obtiene un vector con un par de coordenadas para cada uno de los usuarios generados. Se ha decidido que, para facilitar la implementación, cada usuario tendrá un solo flujo por lo que, si hay N

usuarios, se tendrá un total de  $N$  flujos (unos ascendentes y otros descendentes).

Con este vector posteriormente se calcula la matriz de coberturas y la asociación al AP más cercano.

### 4.3 Cálculo de matriz de cobertura

El siguiente paso será conocer a qué AP se puede conectar cada uno de los usuarios ya que dependiendo de la ubicación de ambos (AP y usuario), se podrá decodificar de manera correcta la información transmitida o no. Para ello se realiza el cálculo de la matriz de coberturas. Suponiendo que el conjunto de AP es

$$A = \{A_1, \dots, A_n\}$$

y el conjunto de usuarios es:

$$S = \{S_1, \dots, S_j\}$$

el primer paso es calcular a qué AP se puede conectar cada usuario.

Tal como se ha descrito en el apartado 3.1, en un sistema real cada AP debería monitorizar las transmisiones de los usuarios en las distintas bandas frecuenciales para determinar a qué usuarios podría dar servicio (aquellos de los que recibe información). Sin embargo, para modelar este procedimiento a la hora de simular, es suficiente con conocer si un usuario tiene cobertura con un determinado AP a partir de la distancia a la que se encuentran y los parámetros de transmisión utilizados en el simulador.

En primer lugar, se han consultado los detalles del modelo de propagación empleado en la documentación de ns-3 [18]. El modelo que se va a utilizar es el “log distance”, el cual emplea la siguiente expresión para calcular las pérdidas de propagación:

$$L = L_0 + 10 n \log_{10}\left(\frac{d}{d_0}\right)$$

Los parámetros de transmisión de los AP y de los usuarios en el simulador son los siguientes:

*Tabla 4.1: Valores utilizados en ns-3*

Ganancia de transmisión – $TXGain$	1 dB
Ganancia de recepción – $RXGain$	1 dB
Potencia de transmisión – $TXPower$	16.0206 dBm
CCA Detection Threshold – $CCATh$	-99 dBm
Reception Threshold – $RXTh$	-96 dBm

con:

$$n = 3$$

$$L_0 = 46,6777 \text{ dB}$$

$$d_0 = 1 \text{ m}$$

A partir de estos valores se puede calcular la distancia máxima de recepción. La expresión que relaciona las potencias de transmisión y recepción es:

$$TXPower + TXGain - L + RXGain \geq RXTh$$

Con lo que el umbral de las pérdidas de propagación para que haya interferencia es:

$$16,0286 \text{ dBm} + 1 \text{ dB} - L + 1 \text{ dB} \geq -96 \text{ dBm}$$

$$L \leq 114,0286 \text{ dB}$$

Si ahora se sustituye en la expresión de las pérdidas, se tiene:

$$114,0286 = 46,6777 + 10 * 3 * \log_{10}(d)$$

$$\log_{10}(d) = \frac{114,0286 - 46,6777}{30} = 2,24503$$

$$d = 10^{2,24503} = 175,804 \text{ m}$$

Por lo tanto, la distancia máxima de cobertura obtenida es 175,804 metros.

Una vez conocida la distancia a la que se obtiene cobertura, se puede calcular la matriz de coberturas  $C$ .

$$C = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{j1} & \cdots & C_{jn} \end{bmatrix}$$

$C_{jn} = 1$  si el Cliente " $n$ " tiene cobertura con el AP " $j$ "

Para ello se ha generado un programa escrito en C++ como en el caso anterior.

En el programa se debe indicar el vector de coordenadas de todos los AP, el vector de coordenadas generado en el paso anterior con las posiciones de los usuarios y el número de AP y de usuarios presentes en nuestro escenario. Una vez ejecutado el programa, devuelve la matriz  $C$ .

## 4.4 Cálculo de matriz de interferencia

Una vez calculada la matriz de cobertura, se debe calcular qué AP se interfieren entre sí. Si bien teóricamente para calcular la interferencia

debería utilizarse el umbral  $CCATh$ , en una implementación práctica sólo puede detectarse potencial interferencia entre AP si estos son capaces de decodificar los *beacon* de los otros, por lo que también en este caso se utiliza como distancia de interferencia los 175 metros calculados anteriormente (en lugar de los 221 que se obtendrían con el  $CCATh$ ). En este caso, se emplea el mismo programa utilizado anteriormente, indicando la posición de los AP y comparándolas con las de los propios AP para crear la matriz de interferencias  $I$ .

$$I = \begin{bmatrix} I_{11} & \cdots & I_{1n} \\ \vdots & \ddots & \vdots \\ I_{j1} & \cdots & I_{jn} \end{bmatrix}$$

$$I_{jn} = 1 \text{ si el AP "n" se interfiere con el AP "j"}$$

Una vez ejecutado el programa, devuelve la matriz  $I$ .

## 4.5 Cálculo AP más cercano a cada usuario

Posteriormente se calcula cuál sería el AP al que se asociaría cada usuario en caso de elegir el más cercano. De esta manera se dispone de un escenario principal en el que basarnos para realizar nuestras pruebas.

Para ello se ha desarrollado otro programa al que se le indica, como en el caso del cálculo de la matriz de coberturas, el vector de coordenadas de los usuarios y de los AP. La diferencia es que, en este caso, el programa irá calculando la distancia a la que se encuentra cada uno de los AP con respecto a los usuarios.

Al finalizar la ejecución se obtiene un vector indicando a qué AP tendría que conectarse cada uno de los usuarios en caso de elegir el más cercano. De esta forma se genera el escenario de referencia con el que comparar posteriormente la solución propuesta.

Teniendo la matriz de interferencias, la matriz de coberturas y la asociación por defecto ya solamente falta conocer cómo se puede aproximar la tasa efectiva que le corresponde a cada flujo de manera teórica.

## 4.6 Ejecución del juego

Para la ejecución del juego, se ha desarrollado otro programa más complejo que se encarga de realizar todos los pasos descritos en el apartado 3.2.3.

Para poder ejecutarse, requiere como entada:

- Matriz de cobertura en forma de vector (para transformar la matriz en un vector hay un programa en bash que rápidamente nos lo transforma).
- Matriz de interferencias en forma de vector.
- Canal en el que transmite cada uno de los AP.
- Vector con el AP más cercano a cada uno de los usuarios.
- Vector indicando si el usuario tiene flujo ascendente o flujo descendente.

Con esta información, se ejecuta el programa que realiza el juego definido. Al finalizar indica en qué canal debe transmitir cada AP y a qué AP se debe asociar cada uno de los usuarios.

A continuación, en el siguiente capítulo se evalúan las prestaciones del algoritmo propuesto.

# Capítulo 5

## Evaluación del algoritmo

En este capítulo se van a presentar las diferentes evaluaciones que se han hecho del algoritmo propuesto para comprobar su funcionamiento. Primero se presentan los parámetros que se van a emplear en el simulador ns-3 y posteriormente se pasará analizar a los resultados en los diferentes escenarios.



## 5.1 Parámetros ns-3

Los parámetros que se van a utilizar en ns-3 para simular los distintos escenarios son los siguientes:

*Tabla 5.1: Parámetros Wi-Fi:*

Tasa de transmisión	6 Mbps
Modelo de transmisión	Log-distance path loss model
Estándar utilizado	IEEE 802.11-2012

Parámetros de transmisión de los AP y de los equipos de usuario (ver Tabla 4.1 en el apartado 4.3).

*Tabla 5.2: Parámetros de flujo de transmisión:*

Tráfico TCP	
<b>Tamaño de paquete</b>	1500 bytes

Como se ha comentado previamente, la tasa de transmisión en Wi-Fi es variable y se negocia entre los usuarios y el AP. Sin embargo, esa negociación no está implementada en ns-3, por lo que por motivos de complejidad no se ha incluido la posibilidad de que existan tasas variables. Así, se ha definido una tasa fija de 6 Mbps para todos los flujos. Como posible línea futura se podría mejorar la programación en ns-3 para incluir esta tasa negociada dependiendo de la distancia a la que se encuentren los usuarios.

A continuación, se muestra la validación del método de estimación del *throughput* presentado en la sección 3.1.1.

## 5.2 Evaluación del método de estimación del *throughput*

Para comprobar la validez del método de estimación del *throughput* se han simulado varios escenarios en ns-3 con distintos patrones de interferencia y número de usuarios asociados por AP. En la figura 5.1 se observa el primero de ellos. Una línea roja uniendo dos AP indica que dichos AP se interfieren, mientras que el número encima de cada AP indica el número de usuarios que hay por AP generando tráfico ascendente. La duración total de la simulación es de 100 segundos y representamos el *throughput* medio en esos 100 segundos.

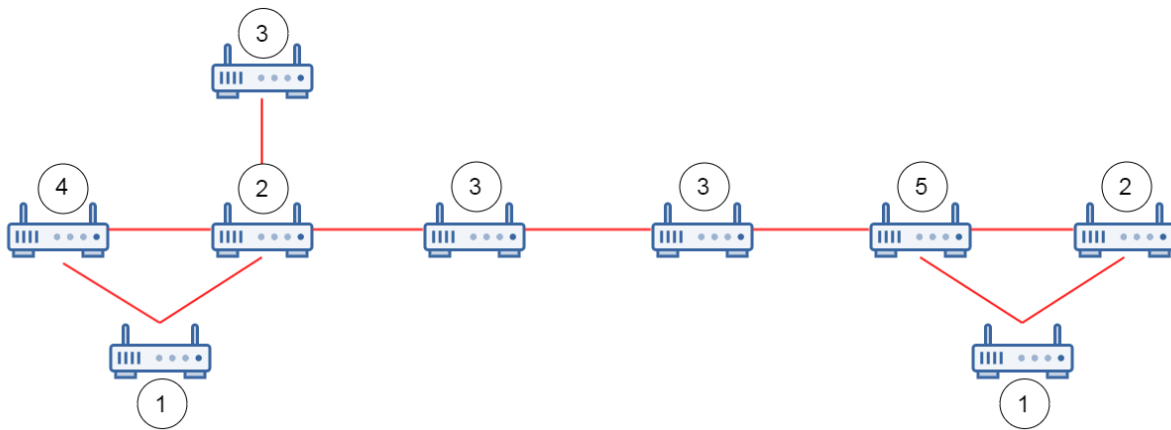


Figura 5.1: Escenario número 1 en el que probar el cálculo teórico.

Tras ejecutar el algoritmo método de estimación del *throughput* en el escenario anterior y simularlo también en ns-3, se obtiene que el *throughput* total esperado según la estimación es de 15,55 Mbps, mientras que la obtenida con ns-3 es de 15,46 Mbps, por lo que la estimación realizada, medida de forma global, es buena. A continuación, se presentan los resultados desglosados flujo a flujo. Para cada fila, la primera dirección IP corresponde a la de la estación de usuario y la segunda a la del AP al que está conectado.

*Tabla 5.3: Throughput simulado vs throughput calculado teóricamente en el escenario 1.*

Flujo	Throughput simulado (MB/S)	Throughput esperado(MB/s)
10.1.4.1--->10.1.4.5	0,62	0,60
10.1.4.2--->10.1.4.5	0,50	0,60
10.1.4.3--->10.1.4.5	0,33	0,60
10.1.4.4--->10.1.4.5	0,30	0,60
10.1.5.1--->10.1.5.2	1,19	0,60
10.1.6.1--->10.1.6.3	0,13	0,27
10.1.6.2--->10.1.6.3	0,10	0,27
10.1.7.1--->10.1.7.4	0,68	1,01
10.1.7.2--->10.1.7.4	0,56	1,01
10.1.7.3--->10.1.7.4	0,45	1,01
10.1.8.1--->10.1.8.4	0,25	0,68
10.1.8.2--->10.1.8.4	0,20	0,68
10.1.8.3--->10.1.8.4	0,14	0,68
10.1.9.1--->10.1.9.4	1,10	0,32
10.1.9.2--->10.1.9.4	0,91	0,32
10.1.9.3--->10.1.9.4	0,70	0,32
10.1.10.1--->10.1.10.6	0,13	0,32
10.1.10.2--->10.1.10.6	0,09	0,32
10.1.10.3--->10.1.10.6	0,08	0,32
10.1.10.4--->10.1.10.6	0,06	0,32
10.1.10.5--->10.1.10.6	0,06	0,32
10.1.11.1--->10.1.11.3	1,72	0,98
10.1.11.2--->10.1.11.3	1,38	0,98
10.1.11.2--->10.1.11.3	0,21	0,49

Como se puede observar, aunque el valor total del *throughput* se aproxima bastante al estimado, el valor individual de cada uno de los flujos en algunos casos es bastante alejado del estimado.

Tras realizar varias simulaciones se pudo observar que, en aquellos escenarios en los que exista un AP que esté rodeado por varios AP con los que se interfiere, el método de estimación propuesto tiende a sobreestimar su capacidad. Esto provoca dos cosas, en primer lugar, el *throughput* real de ese AP es mucho menor que el esperado y, en segundo lugar, ese *throughput* que en la simulación se detrae de ese AP, se reparte entre el resto de los AP con los que tiene interferencias. Por eso en algunos flujos se obtiene más *throughput* del esperado y en otros cercanos a estos ocurre justo lo contrario. Por otro lado, en el método de estimación propuesto, se ha supuesto que el tiempo de transmisión disponible en un AP se reparte de modo equitativo entre todos los usuarios conectados, cuando en las simulaciones se ve que en algunos casos no es así. Esto se debe a que para calcular teóricamente el tiempo de transmisión que le toca a cada estación habría que tener también en cuenta el patrón de interferencias a cada usuario (un usuario más interferido por otros usuarios o AP que otro va a tener en la realidad menos tiempo de transmisión, aunque estén en el mismo AP). Sin embargo, es imposible calcular estos patrones de interferencia a priori ya que no se puede tener información de la ubicación de cada usuario en la red.

A continuación, se muestran los resultados en otro escenario:

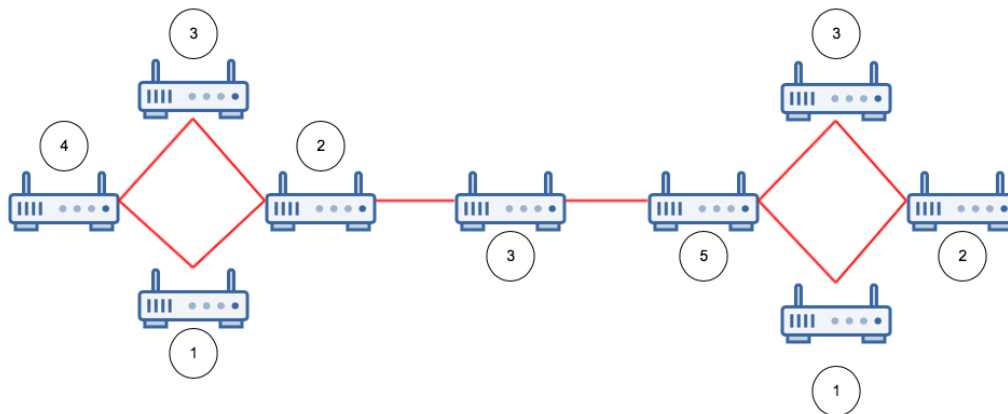


Figura 5.2: Escenario número 2 en el que probar el cálculo teórico.

En la tabla 5.4 mostramos el *throughput* por usuario, mientras que en la tabla 5.5 mostramos el *throughput* agregado por AP.

*Tabla 5.4: Throughput simulado por flujo en el escenario 2.*

Flujo	Throughput simulado (MB/s)	Throughput esperado (MB/s)
10.1.4.1--->10.1.4.5	0,82	0,47
10.1.4.2--->10.1.4.5	0,65	0,47
10.1.4.3--->10.1.4.5	0,49	0,47
10.1.4.4--->10.1.4.5	0,40	0,47
10.1.5.1--->10.1.5.2	0,99	0,87
10.1.6.1--->10.1.6.4	0,59	0,39
10.1.6.2--->10.1.6.4	0,46	0,39
10.1.6.3--->10.1.6.4	0,26	0,39
10.1.7.1--->10.1.7.3	1,37	0,39
10.1.7.2--->10.1.7.3	0,16	0,39
10.1.8.1--->10.1.8.4	0,49	0,93
10.1.8.2--->10.1.8.4	0,39	0,93
10.1.8.3--->10.1.8.4	0,31	0,93
10.1.9.1--->10.1.9.6	0,32	0,39
10.1.9.2--->10.1.9.6	0,55	0,39
10.1.9.3--->10.1.9.6	0,46	0,39
10.1.9.4--->10.1.9.6	0,31	0,39
10.1.9.5--->10.1.9.6	0,35	0,39
10.1.10.1--->10.1.10.2	1,39	0,50
10.1.11.1--->10.1.11.4	0,68	0,36
10.1.11.2--->10.1.11.4	0,45	0,36
10.1.11.3--->10.1.11.4	0,34	0,36
10.1.12.1--->10.1.12.3	1,19	1,00
10.1.12.2--->10.1.12.3	0,97	0,59

*Tabla 5.5: Throughput simulado agregado por AP vs throughput medio esperado por AP en el escenario 2.*

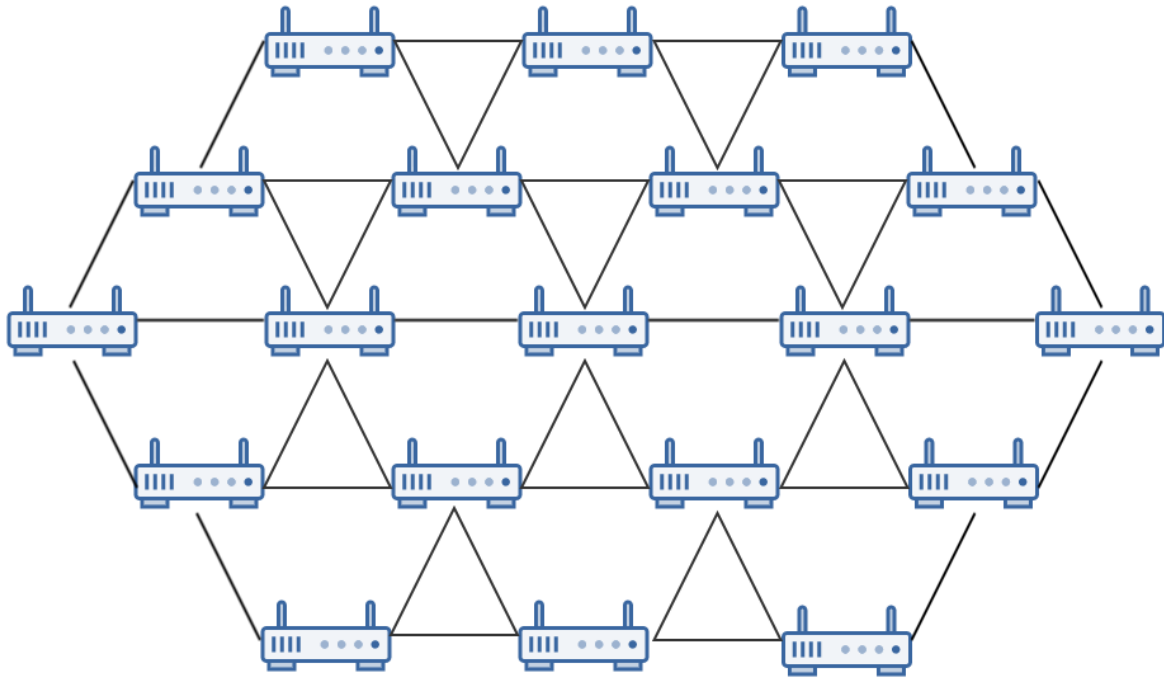
AP	Media del <i>throughput</i> los flujos por AP (MB/s)	<i>Throughput</i> medio esperado (MB/s)
<b>1</b>	0,59	0,47
<b>2</b>	0,99	0,87
<b>3</b>	0,44	0,39
<b>4</b>	0,76	0,39
<b>5</b>	0,59	0,93
<b>6</b>	0,40	0,39
<b>7</b>	1,39	0,50
<b>8</b>	0,49	0,36
<b>9</b>	1,08	1,00

Igual que en el caso anterior, el AP que se encuentra en medio del escenario es el más perjudicado, obteniendo 0,59 Mbps en vez de 0,93 Mbps como se había estimado. Esto provoca que en el resto de flujos aumente un poco la tasa alcanzada frente a la estimada. De hecho, en el AP7, al solo tener un flujo, la tasa obtenida pasa de los 0,5 Mbps estimados a los 1,39 Mbps. En el resto de casos se obtiene también una tasa algo mayor a la estimada, si bien el desvío no es muy pronunciado.

Como línea futura se podría intentar modelar este comportamiento realizando un estudio más exhaustivo de por qué se producen estas variaciones. En cualquier caso, al menos permite realizar una primera aproximación a cómo se realiza el reparto de ancho de banda entre flujos, que sirva como parámetro de entrada al algoritmo de selección de canal frecuencial y asociación de usuarios. En el siguiente apartado se evalúan las prestaciones de dicho algoritmo.

## 5.3 Evaluación del algoritmo de selección de canal frecuencial y asociación de usuarios

Para realizar estas simulaciones, vamos a asumir un escenario en el que hay un administrador que controla los AP. Asumimos que tiene capacidad para decidir en qué ubicación colocar los AP por lo que sería capaz de generar una distribución como la que mostramos a continuación.



*Figura 5.3 Distribución hexagonal de los puntos de acceso.*

Con esa planificación y el reuso frecuencial de los canales 1, 6 y 11 podemos conseguir que, dependiendo de la distancia a la que se encuentren, se cancelen totalmente las interferencias si lo distribuimos como en la imagen siguiente:

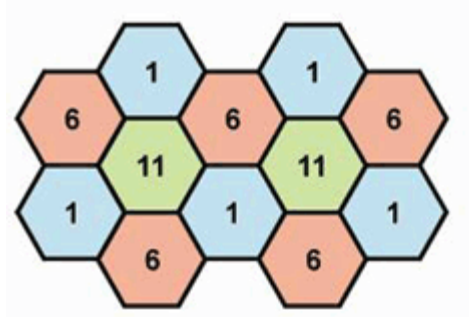


Figura 5.4: Asignación ideal de canal para evitar interferencias en una distribución hexagonal.

No obstante, para aumentar la capacidad de la red, en muchas situaciones se despliega un mayor número de AP, de manera que la distancia entre ellos es menor y puede producirse interferencia entre AP utilizando la misma banda frecuencial. Este es el escenario que vamos a considerar.

También podría resultar interesante evaluar el algoritmo propuesto en un escenario de despliegue aleatorio de AP. No obstante, para ello se debería disponer de un mecanismo de selección de canal inicial por parte de los AP, como el *Least Congested Channel Search* (LCCS) [17], que no está incluido en ns-3. Sin dicho mecanismo, el caso más básico sería comparar el algoritmo propuesto con una selección de canal aleatoria lo cual no resultaría justo (si bien en la práctica, cuando un usuario asigna manualmente el canal de su AP, podría darse dicho escenario). Como línea futura se plantea la implementación del algoritmo LCCS en ns-3 para realizar comparaciones en escenarios de despliegue aleatorios.

Finalmente, el escenario de trabajo es el dado por la figura 5.3, con una distancia mínima entre AP de 87,5 metros. Esto hace que el AP central se vea interferido e interfiera al resto de AP de la red. En definitiva, se produce interferencia entre AP que se encuentran a una distancia de 2 AP.

Sobre esta topología, se va a ir variando el número de usuarios y la ratio de flujos ascendentes y descendentes que hay. Por ejemplo, si tenemos 20 usuarios y la ratio es 3 ascendentes por cada descendente, tendríamos 15



flujos ascendentes en 15 usuarios distintos y 5 flujos descendentes a los otros cinco usuarios (Como los flujos descendentes se transmiten por el AP, la tasa que obtiene este se tiene que dividir entre 5 para saber cuál es la que le corresponde a cada flujo). Para cada escenario, se va a comparar tres soluciones: la primera es la solución por defecto, y corresponde a la asignación frecuencial realizada en la Figura 5.4. con cada usuario conectado al AP más cercano (defecto en las figuras). En la segunda solución se intenta maximizar el *throughput* total de la red haciendo que la función de utilidad del juego propuesto sea la suma de las tasas que consigue cada flujo (max-cap en las figuras). La tercera solución corresponde a la maximización de la suma de los logaritmos de las tasas de cada flujo (max-log en las figuras). Igual que en la anterior, esta maximización se busca haciendo que la función de utilidad en el juego sea esta suma.

A continuación, pasamos a explicar los resultados obtenidos medidos en términos del índice de Jain, del *throughput* total y de la CDF (Cumulative Distribution Function) de las tasas de los flujos de la red.

Para extraer estos resultados se han promediado varias simulaciones (dependiendo del tiempo que le costara a cada una se han realizado más o menos simulaciones) de 100 segundos cada una.

### 5.3.1 Índice de Jain

Para evaluar cuantitativamente la equidad del reparto de recursos, utilizamos el índice de Jain que viene dado por la siguiente expresión:

$$f(x) = \frac{(\sum c_i)^2}{N \sum c_i^2}$$

Donde  $c_i$  es igual a la tasa efectiva disponible en el flujo  $i$  y  $N$  es el número total de flujos en la red.

A continuación, se muestran en las Figuras 5.6-5.8 los resultados obtenidos en función del número de usuarios y según la ratio de flujos ascendentes y descendentes. Se puede observar como en todos los casos el mejor índice de Jain se obtiene cuando intentamos maximizar la suma del logaritmo de las capacidades, independientemente del número de usuarios y de la proporción que tengamos entre flujos ascendentes y descendentes. Esto se debe a que esta maximización tiende a repartir los recursos disponibles de un modo más equitativo que la solución por defecto o cuando se maximiza el *throughput*. De hecho, cuando se maximiza el *throughput*, no se tiene en cuenta la equidad en el reparto de los recursos, por lo que los resultados obtenidos son similares al caso por defecto. En efecto, al intentar maximizar el *throughput* total de la red, lo que se busca es aislar los AP de manera que haya las menores interferencias posibles y, si no es capaz de aislarlos, asocia a los usuarios sin importar el *throughput* que den (puede juntar 10 usuarios en un AP y 1 usuario en otro AP contiguo sin importarle que el reparto sea más injusto ya que el *throughput* es el mismo).

Cuanto mayor es la proporción de flujos ascendentes mayor será la flexibilidad que se tendrá a la hora de recolocar a los usuarios ya que cada flujo ascendente se trata de manera independiente mientras que los flujos descendentes que se ubican en el mismo AP se agrupan en uno solo a la hora de intentar transmitir. Por esta razón, la mejora en el índice de Jain cuando se maximiza el logaritmo es mayor cuanto mayor es el porcentaje de flujos ascendentes.

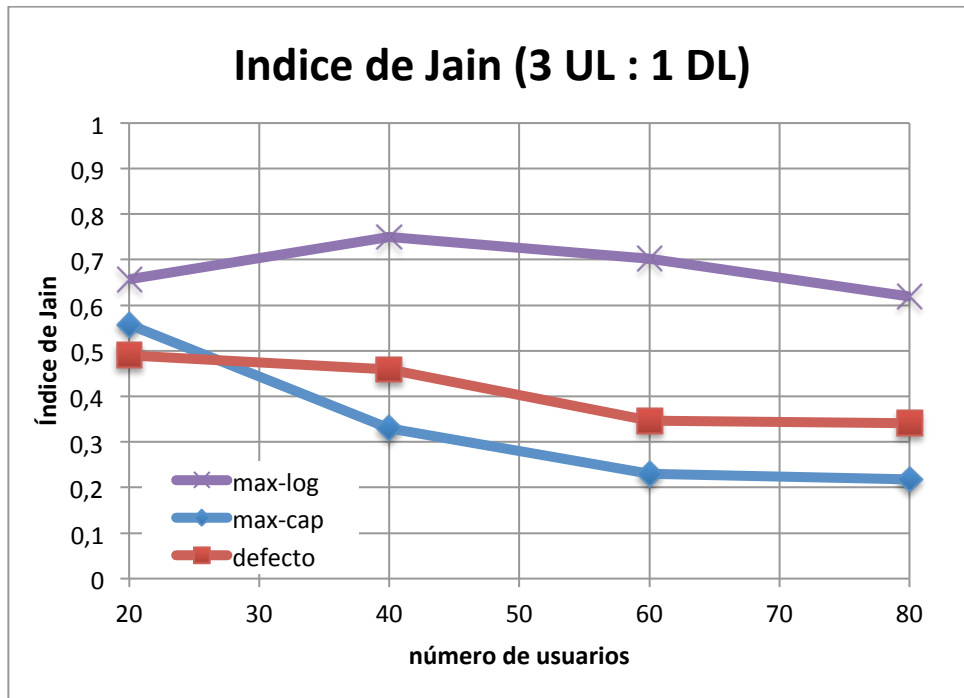


Figura 5.5: Índice de Jain en función del número de usuarios con el triple de flujos ascendentes que descendentes.

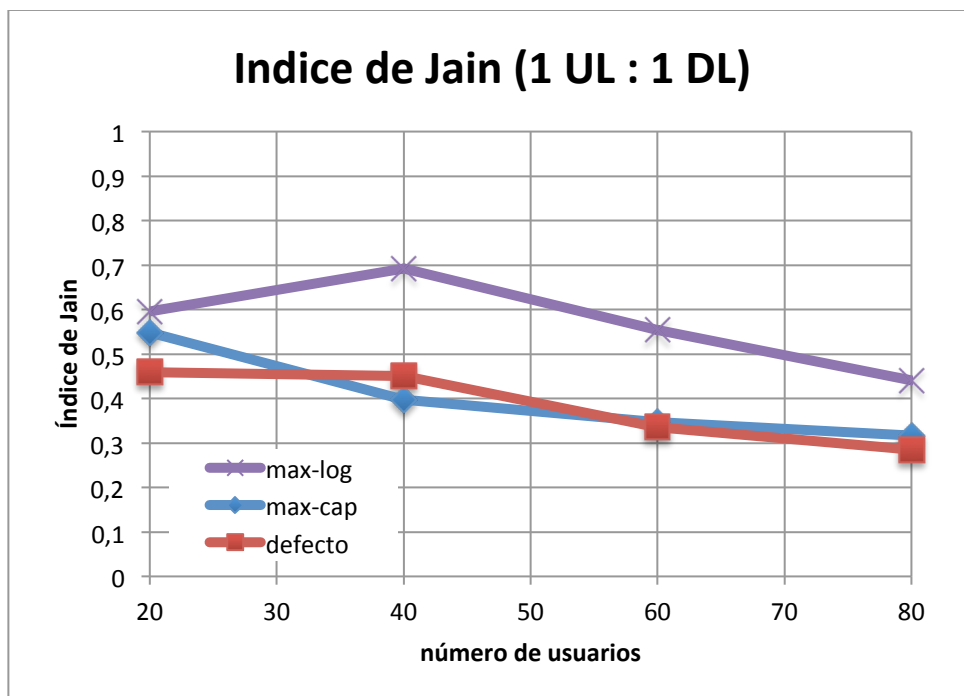


Figura 5.6: Índice de Jain en función del número de usuarios con el mismo número de flujos ascendentes y descendentes.

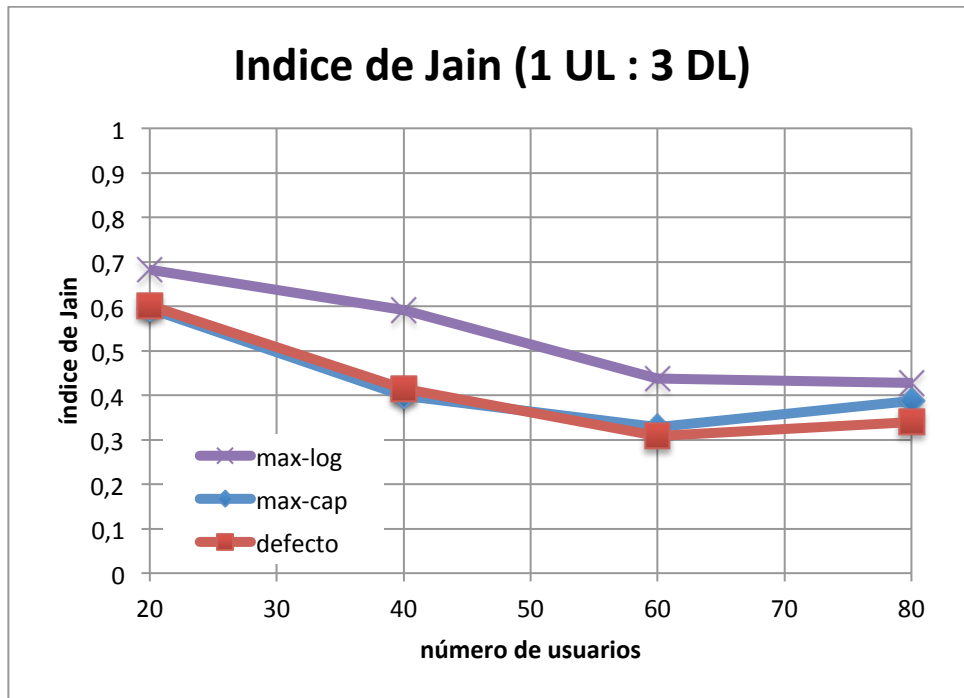


Figura 5.7: Índice de Jain en función del número de usuarios con el triple de flujos descendentes que ascendentes.

### 5.3.3 Suma total del *throughput*

En este apartado se va a comparar el valor total de *throughput* que se puede conseguir en los mismos escenarios del apartado anterior. Como se puede observar en las figuras 5.9-5.11, la solución por defecto es la peor y los resultados obtenidos son muy similares cuando intentamos maximizar el *throughput* o la suma de los logaritmos de las capacidades. Por otro lado, conforme aumenta el número de usuarios, los resultados tienden a parecerse más ya que al final todos los AP acaban teniendo usuarios asociados y, por tanto, todos los AP pueden entregar *throughput* (cuando hay pocos, puede darse el caso de que haya AP sin usuarios en el caso por defecto).

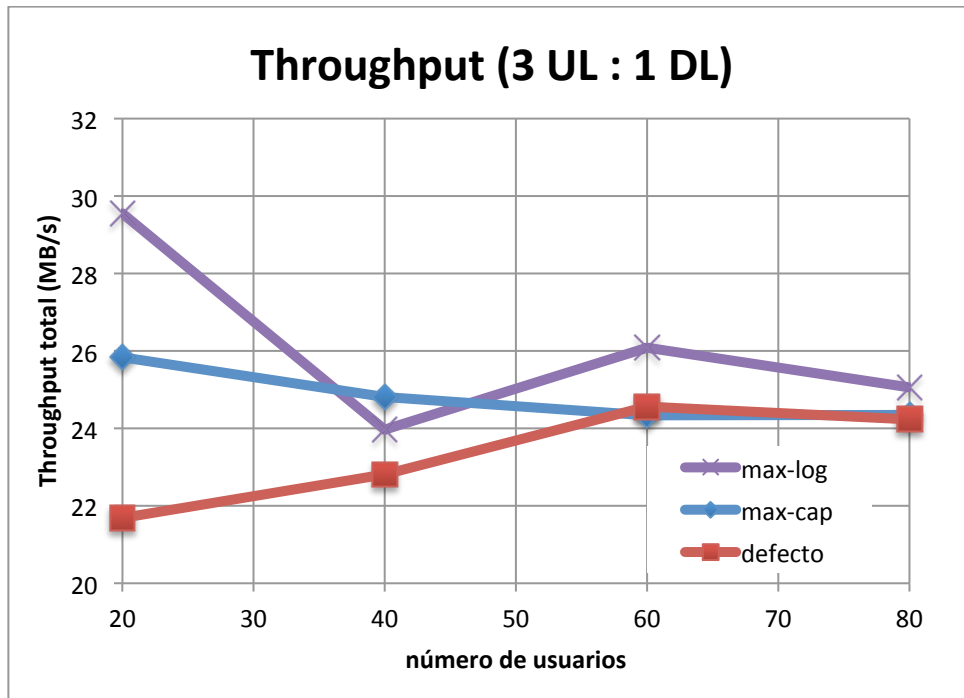


Figura 5.8: Throughput total en función del número del número de usuarios con el triple de flujos ascendentes que descendentes.

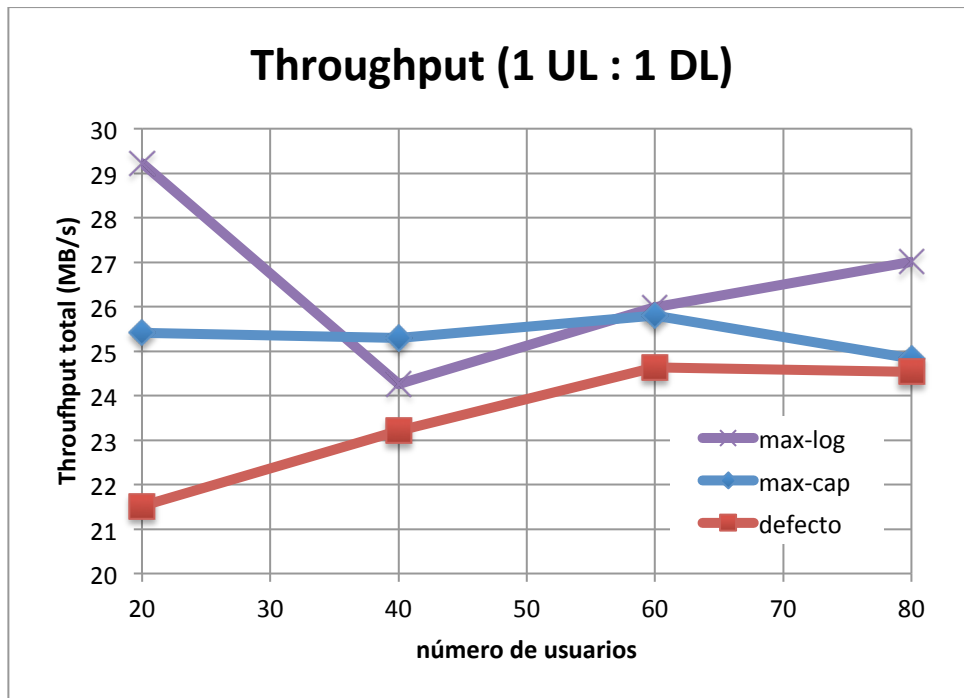


Figura 5.9: Throughput total en función del número de usuarios con el mismo número de flujos ascendentes y descendentes.

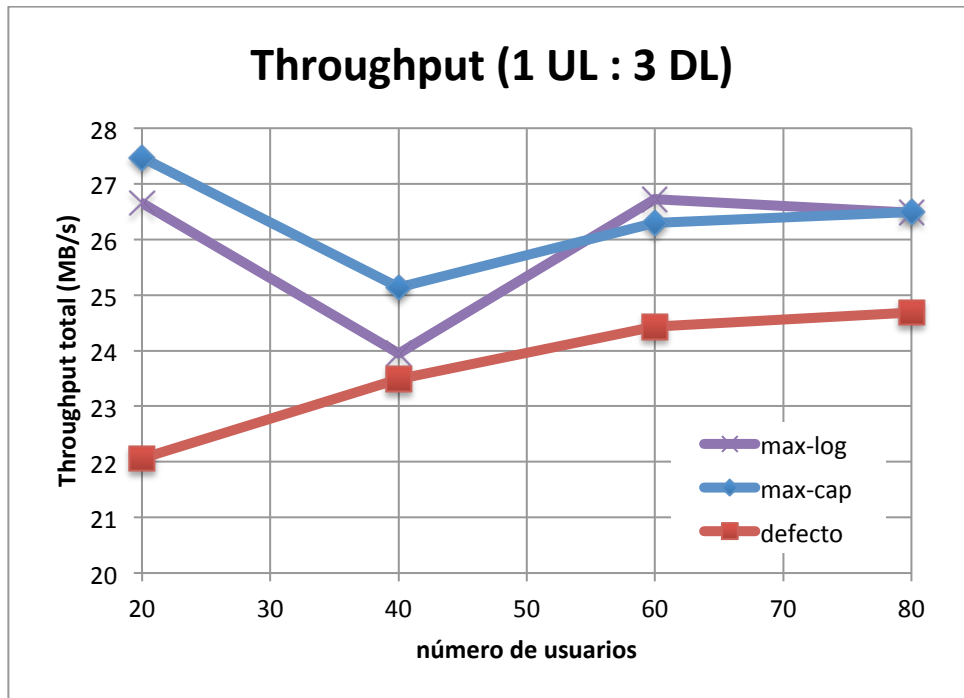


Figura 5.10: Throughput total en función del número de usuarios con el triple de flujos descendentes que ascendentes.

### 5.3.4 Función de distribución acumulada

Finalmente, se analiza la función de distribución acumulada (CDF) obtenida con las distintas soluciones, las cuales nos indican la probabilidad de que la capacidad obtenida por un flujo esté por debajo de un valor dado. Estas distribuciones son muy útiles ya que permiten ver cómo está repartido el ancho de banda entre los distintos flujos. En las figuras 5.12-5.14 se representan las CDF para distintas ratios de flujos ascendentes/descendentes cuando hay 20 usuarios, en las figuras 5-15-5.17, cuando hay 40 usuarios, en las figuras 5.18-5.20 cuando hay 60 usuarios y las figuras 5.21-5.23 cuando hay 60 usuarios.

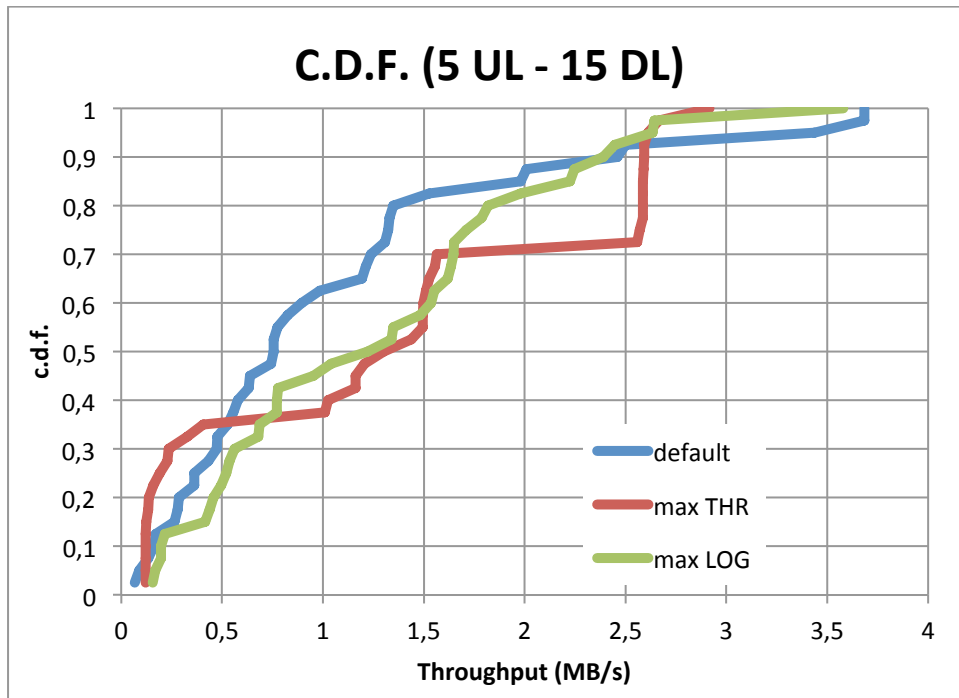


Figura 5.11: C.D.F con 5 flujos ascendentes y 15 descendentes.

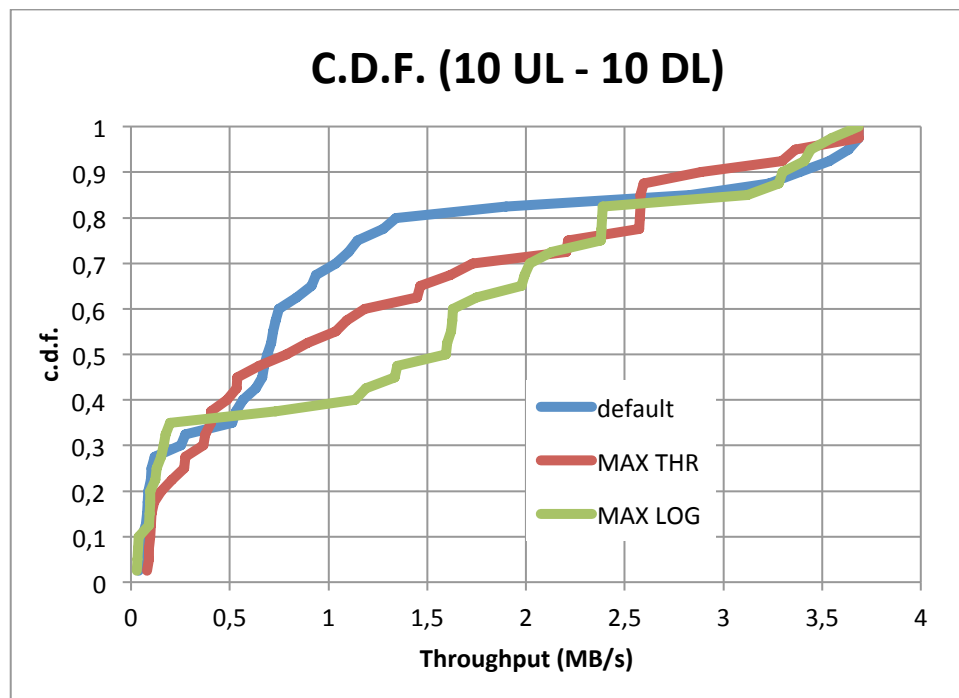


Figura 5.12: C.D.F con 10 flujos ascendentes y 10 descendentes.

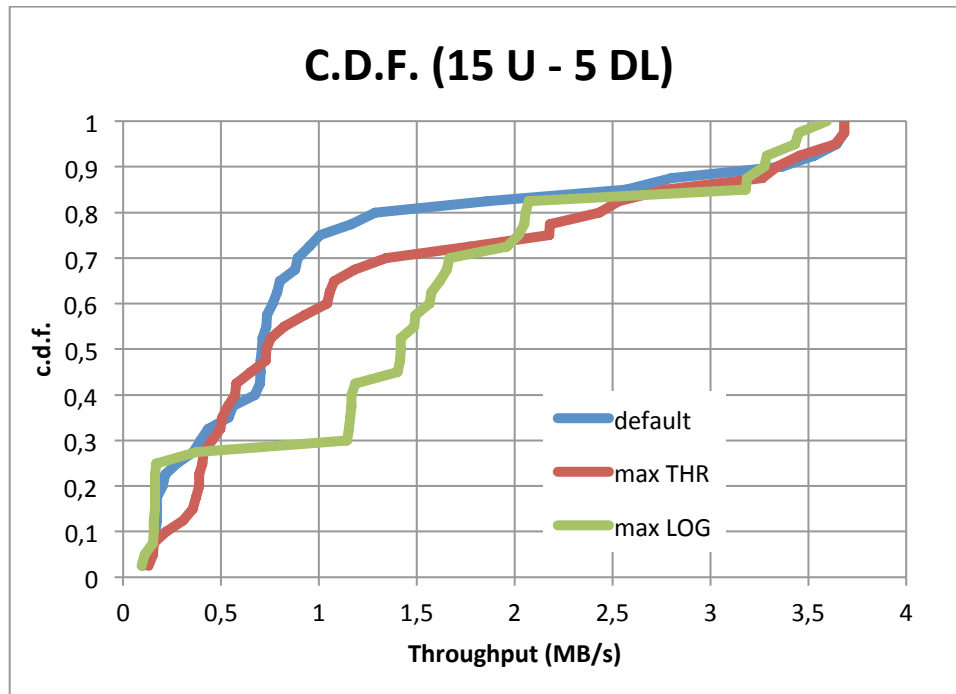


Figura 5.13: C.D.F con 15 flujos ascendentes y 5 descendentes.

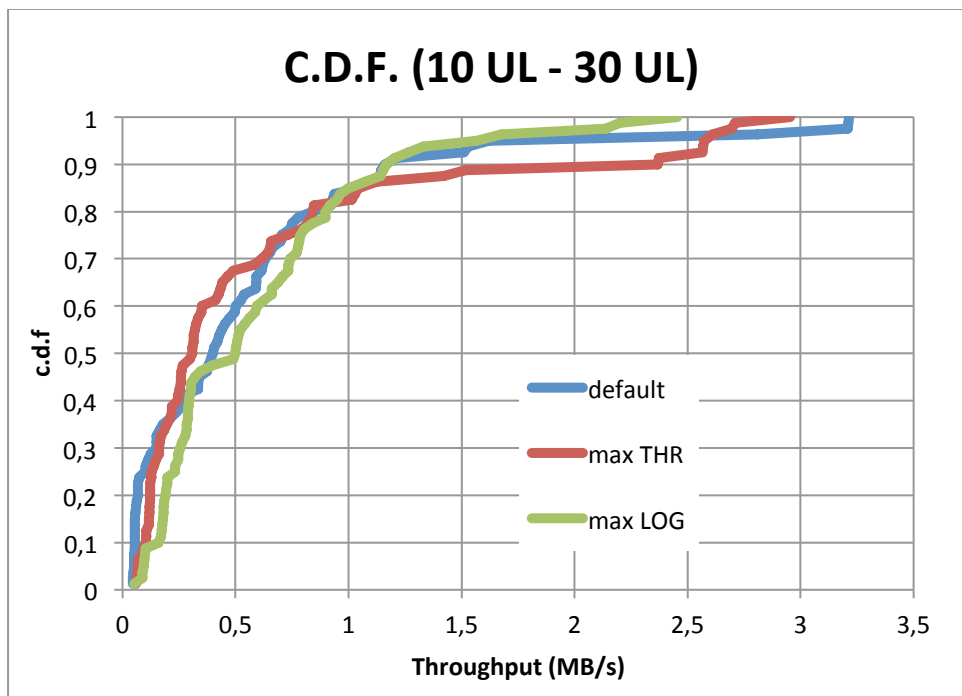


Figura 5.14: C.D.F con 10 flujos ascendentes y 30 descendentes.



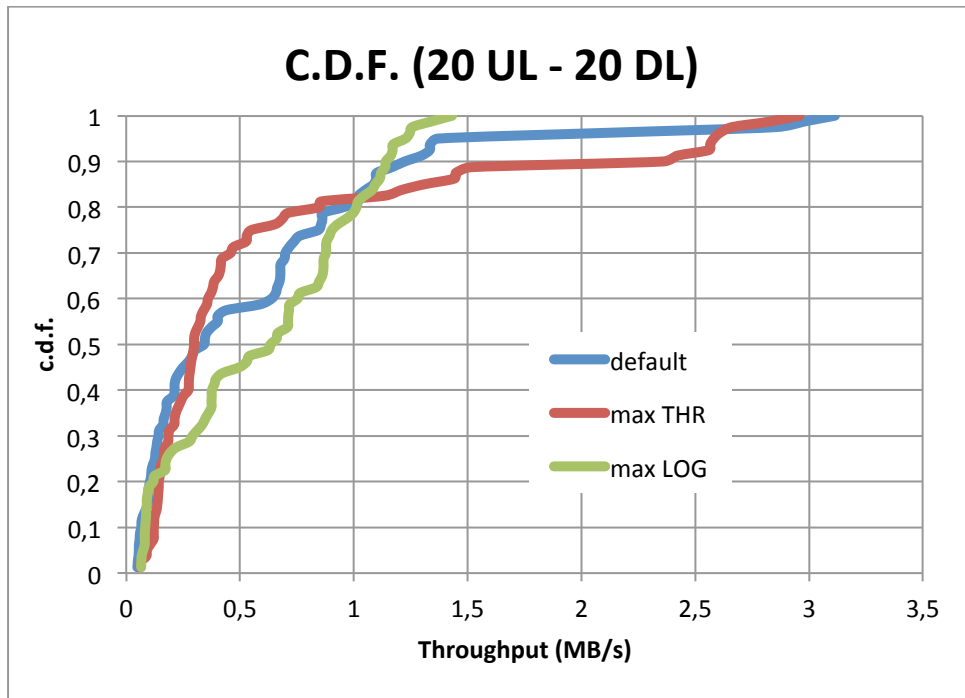


Figura 5.15: C.D.F con 20 flujos ascendentes y 20 descendentes.

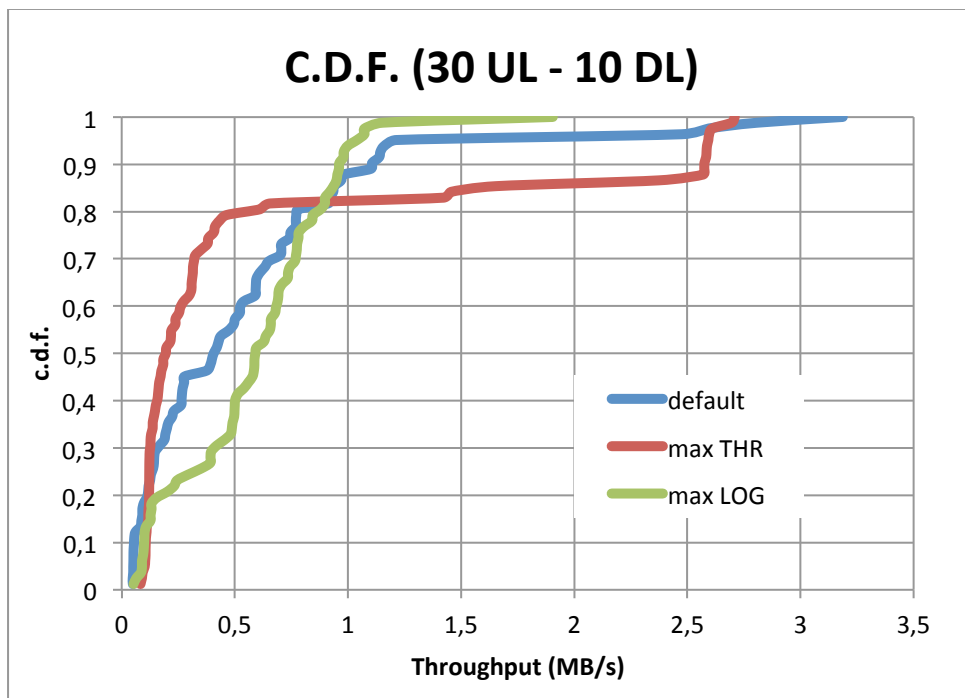


Figura 5.16: C.D.F con 30 flujos ascendentes y 10 descendentes.

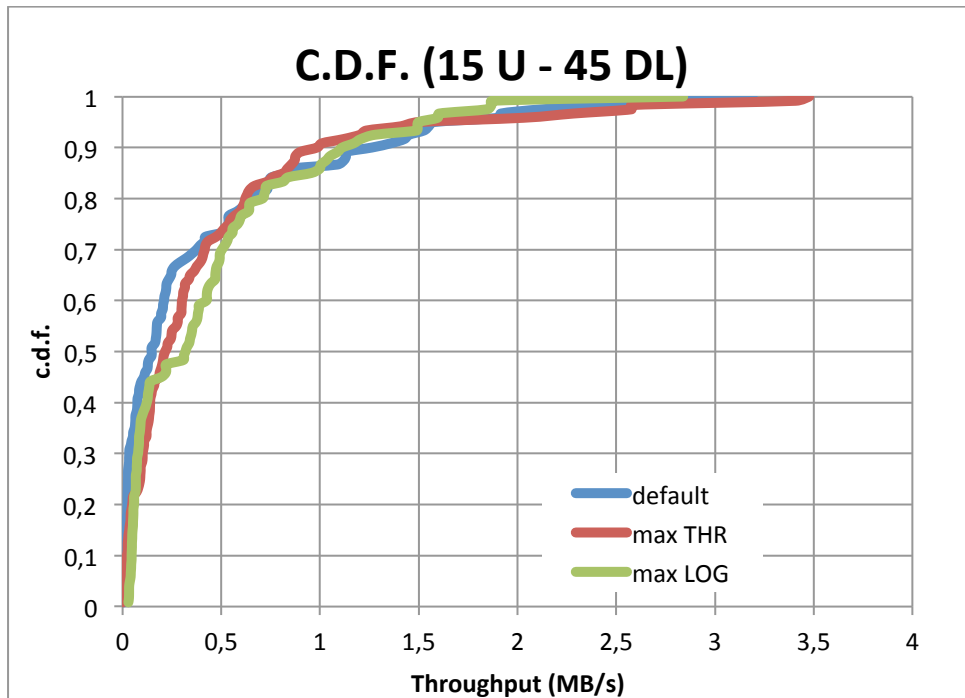


Figura 5.17: C.D.F con 15 flujos ascendentes y 45 descendentes.

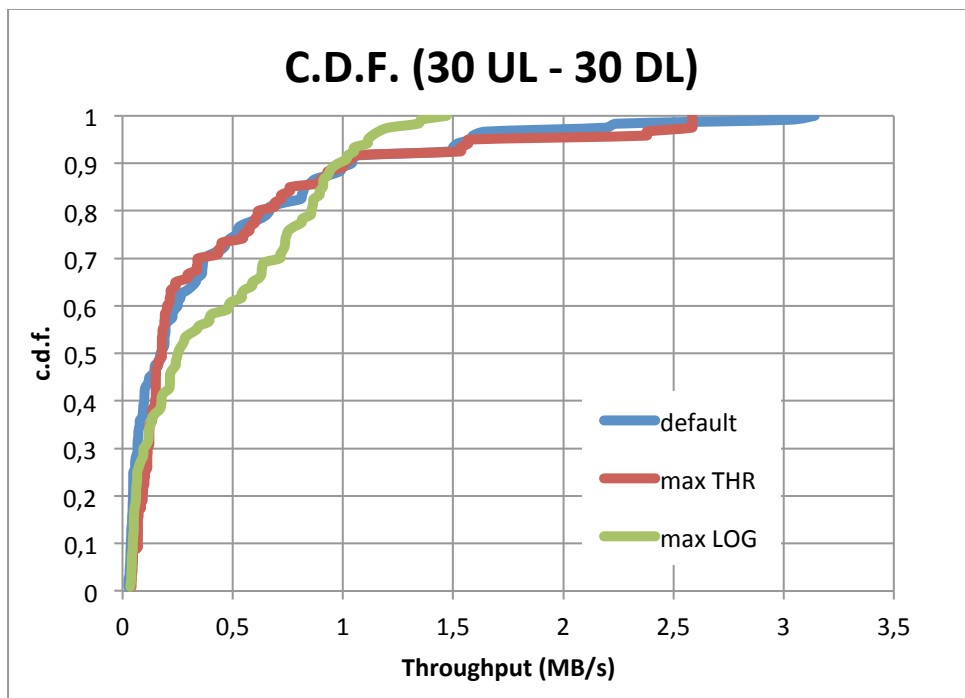


Figura 5.18: C.D.F con 30 flujos ascendentes y 30 descendentes.

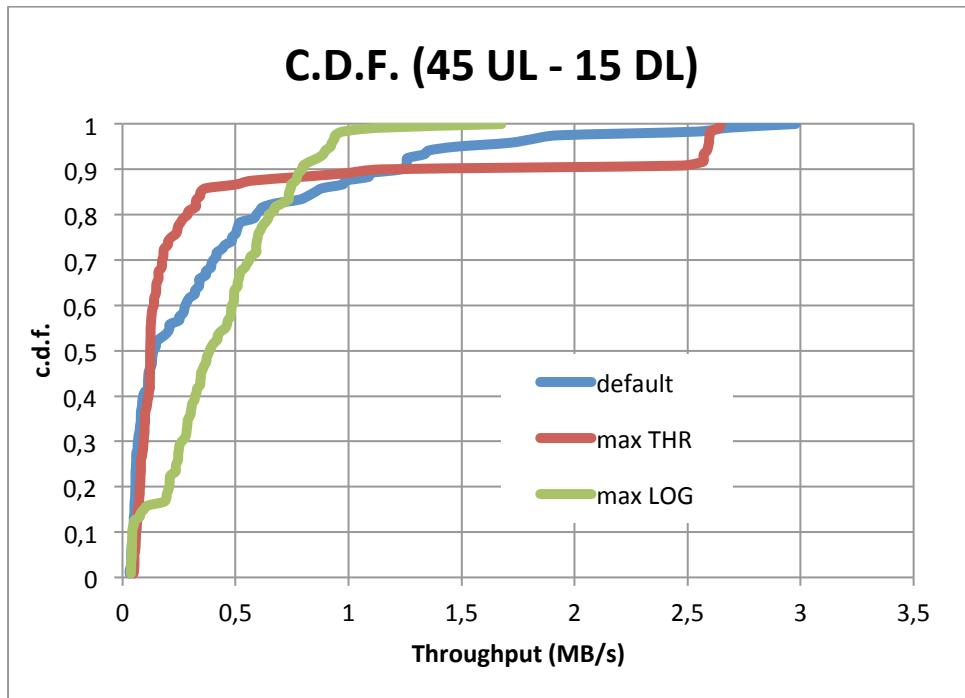


Figura 5.19: C.D.F con 45 flujos ascendentes y 15 descendentes.

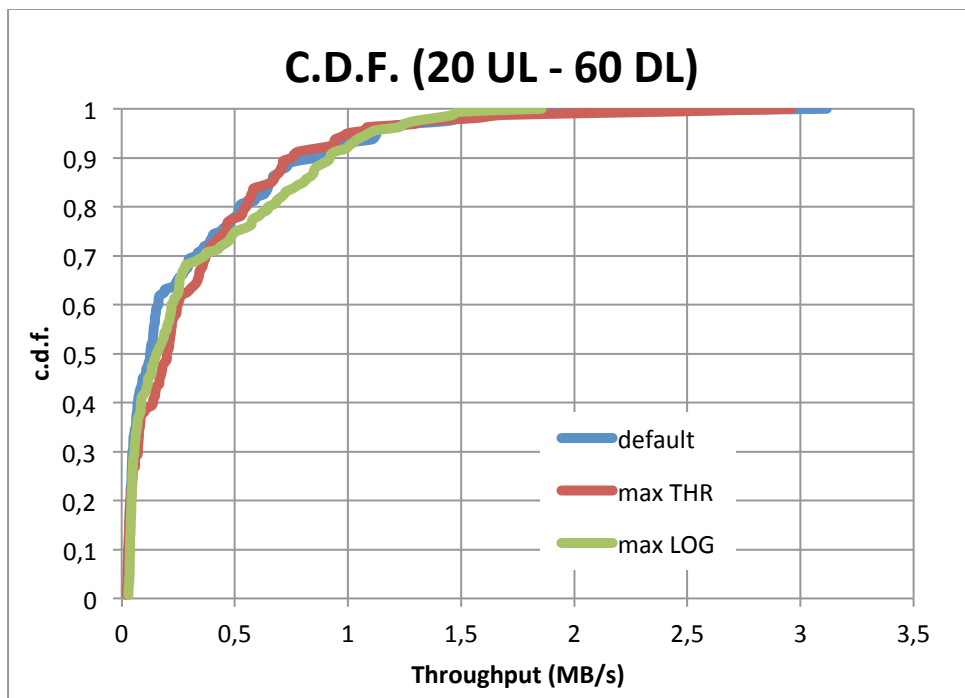


Figura 5.20: C.D.F con 20 flujos ascendentes y 60 descendentes.

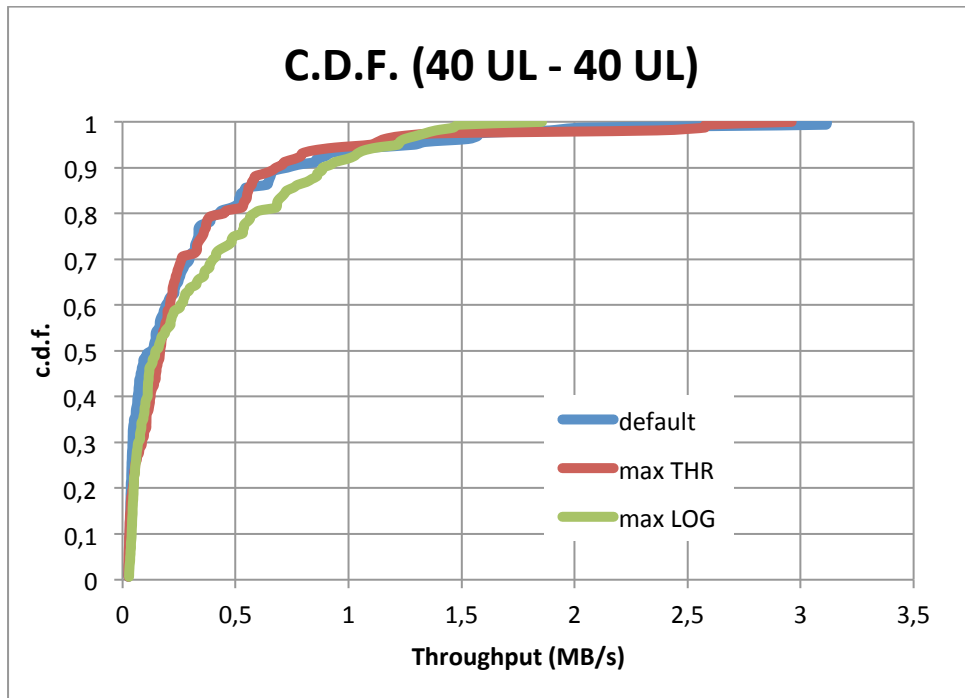


Figura 5.21: C.D.F con 40 flujos ascendentes y 40 descendentes.

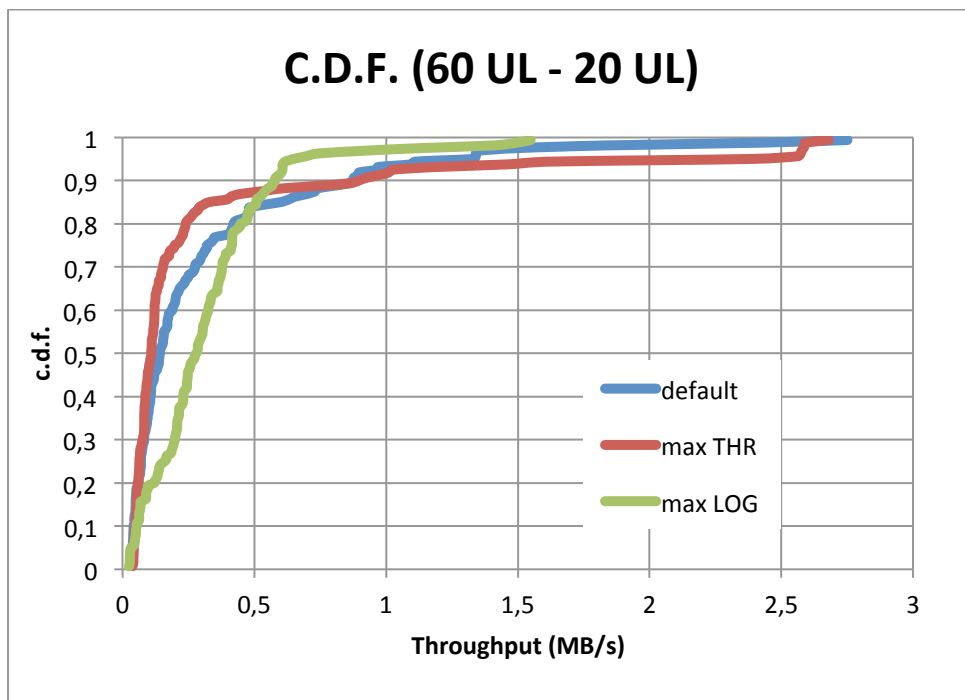


Figura 5.22: C.D.F con 60 flujos ascendentes y 20 descendentes.

Como se puede observar en todas las CDF, cuanto mayor es la proporción de flujos ascendentes que descendentes, mejor reparto conseguimos con el

algoritmo en el que intentamos maximizar el logaritmo. Como se ha comentado antes, esto se debe a que tenemos mayor libertad a la hora de recolocar a los flujos ascendentes que a los descendentes. También se puede ver que conforme aumenta el número de usuarios, la diferencia entre los diferentes algoritmos se va estrechando. Esto se debe a que conforme se va saturando la red, todas los AP tienden a tener el mismo número de usuarios y la asignación frecuencial tiende a asemejarse a la empleada en el escenario por defecto.

# Capítulo 6

## Conclusiones y líneas futuras de trabajo

### 6.1 Conclusiones

En este trabajo se ha propuesto un algoritmo centralizado para realizar la asignación de canales y asociación de usuarios en una red IEEE 802.11. El algoritmo propuesto mejora, dentro de una distribución hexagonal, el escenario por defecto en el que los AP tienen un canal asignado fijo y los usuarios se conectan al AP más cercano.

El algoritmo propuesto se basa en un sistema centralizado el cual, en base a la ubicación de los AP y de los usuarios, decide en qué canal debe transmitir cada AP y a qué AP se debe asociar cada uno de los usuarios. Este algoritmo se basa en la aplicación de un juego potencial, el cual usa como función potencial el *throughput* total o la suma de los logaritmos de las tasas de cada flujo.

El algoritmo propuesto se ha simulado utilizando la herramienta ns-3, y sus resultados se han evaluado en varios escenarios con diferente número de usuarios y con diferente relación entre flujos ascendentes y descendentes donde se ha comprobado el mejor comportamiento del mismo frente al escenario por defecto en el que no se tenía en cuenta el estado de la red para decidir en qué canal transmite cada AP ni a qué AP se asocian los usuarios.

## 6.2 Líneas futuras

Partiendo de este trabajo, se proponen las siguientes líneas futuras de trabajo:

- Implementar el algoritmo LCCS para tener más escenarios en los que comparar el algoritmo propuesto.
- Implementación del sistema en dispositivos reales, programando e incorporando el algoritmo en los *router* y en un controlador central, de tal modo que se pueda evaluar nuestro sistema. Este trabajo lleva asociado el diseño del protocolo de señalización entre los AP y el controlador.
- Inclusión de tráfico UDP y su consiguiente estimación teórica.
- Integrar el algoritmo propuesto en ns-3 para poder realizar cambios de manera dinámica conforme vaya cambiando el escenario
- Implementar en ns-3 la tasa negociada entre los AP y los usuarios sin tener que definirla manualmente.
- Mejorar la estimación teórica del reparto del medio.

# Bibliografía

- [1] Enlace a la página Web del proyecto Wi-5: <http://www.wi5.eu/> [Fecha de consulta: 15/11/17]
- [2] Enlace a la página Web del programa de investigación “Horizon 2020”: <https://ec.europa.eu/programmes/horizon2020/> [Fecha de consulta: 15/11/17]
- [3] IEEE 802.11-2012: <https://standards.ieee.org/findstds/standard/802.11-2012.html> [Fecha de consulta: 23/11/17]
- [4] Enlace a la página Web de ns-3: <https://www.nsnam.org> [Fecha de consulta: 15/11/17]
- [5] M. Achanta, “Method and Apparatus for Least Congested Channel Scan for Wireless Access Points,” US Patent No. 20060072602, Apr. 2006.
- [6] K. Zhou, X. Jia, L. Xie, Y. Chang, and X. Tang, “Channel Assignment for WLAN by Considering Overlapping Channels in SINR Interference Model”, International Conference on Computing, Networking and Communications (ICNC), Maui, Hawaii, USA 30 Jan.- 2 Feb. 2012.
- [7] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs." SIGMOBILE Mob. Comput. Commun. Rev. 9(3): 19-31, 2005.
- [8] Bondy, J.A., Murty, U.S.R., Graph Theory, Springer, ISBN 978-1-84628-969-9, 2008.



- [9] H. Lee, S. Kim, O. Lee, S. Choi, and S.-J. Lee, “Available bandwidth-based association in IEEE 802.11 wireless LANs,” in Proc. Int. Symp. Model. Anal. Simulat. Wireless Mobile Syst. (MSWiM), Vancouver, BC, Canada, Oct. 2008, pp. 132–139.
- [10] K. Sood, S. Liu, S. Yu, and Y. Xiang, “Dynamic access point association using software defined networking,” in Proc. Int. Telecommun. Netw. Appl. Conf. (ITNAC), Sydney, NSW, Australia, Nov. 2015, pp. 226–231.
- [11] X. Chen, W. Yuan, W. Cheng, W. Liu, and H. Leung, “Access point selection under QoS requirements in variable channel-width WLANs,” IEEE Wireless Commun. Lett., vol. 2, no. 1, pp. 114–117, Feb. 2013.
- [12] M. Liyanage, J. Chirkova, and A. Gurtov, “Access point selection game for mobile wireless users,” in Proc. Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM), Sydney, NSW, Australia, Jun. 2014, pp. 1–6.
- [13] J. B. Ernst, S. Kremer, and J. J. P. C. Rodrigues, “A utility based access point selection method for IEEE 802.11 wireless networks with enhanced quality of experience,” in Proc. IEEE Int. Conf. Commun. (ICC), Sydney, NSW, Australia, Jun. 2014, pp. 2363–2368.
- [14] F. Kelly, Charging and rate control for elastic traffic, Eur. Trans. Telecommun. 8 (1) (1997) 33–37.
- [15] M. J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, 1994.
- [16] D. Monderer, L.S. Shapley, Potential games, Games Econom. Behav. 14 (1996) 124–143.
- [17] M. Achanta. Method and apparatus for least congested channel scan for wireless access points. Apr. 2006.

[18] Enlace al modelo de propagación por defecto utilizado por ns-3:  
[https://www.nsnam.org/docs/release/3.3/doxygen/classns3\\_1\\_1\\_log\\_distance\\_propagation\\_loss\\_model.html](https://www.nsnam.org/docs/release/3.3/doxygen/classns3_1_1_log_distance_propagation_loss_model.html). [Fecha de consulta: 15/11/17]

# **ANEXOS**

# Anexo A

## Simulador de red ns-3

En este anexo se va a explicar el simulador de red ns-3, en el que se han programado los escenarios que se han probado en este trabajo.

Se realizará una breve introducción a ns-3 y, tras esto, se analizarán sus características principales y se explicará cuál es su estructura, revisando los módulos utilizados para el desarrollo de las simulaciones realizadas en este trabajo.

### A.1 Introducción a ns-3

El simulador ns-3 es un simulador de redes basado en eventos discretos, que se emplea principalmente para investigación y educación. El proyecto ns-3 se inició en el año 2006, y es un proyecto de desarrollo de código abierto bajo licencia GNU GPLv2.

ns-3 proporciona una plataforma de simulación implementada principalmente en C++, aunque algunas de sus estructuras están escritas en Python. Es compatible con Linux, Mac OS y FreeBSD.

Este simulador proporciona modelos para trabajar con paquetes de datos y redes, además de incorporar modelos de movilidad y propagación. Todo ello forma un motor de simulación en el se pueden programar simulaciones, de forma que se pueden realizar estudios sobre el comportamiento de los sistemas en un entorno simulado.

Una de sus características más importantes y que más se ha utilizado en este trabajo es el hecho de permitir la definición de nuevas topologías, escenarios y modelos de red, que pueden ser fácilmente integrados y

depurados en el simulador. Se organiza en módulos, ya que de esta forma se facilita la independencia entre los diferentes niveles de la arquitectura de ns-3.

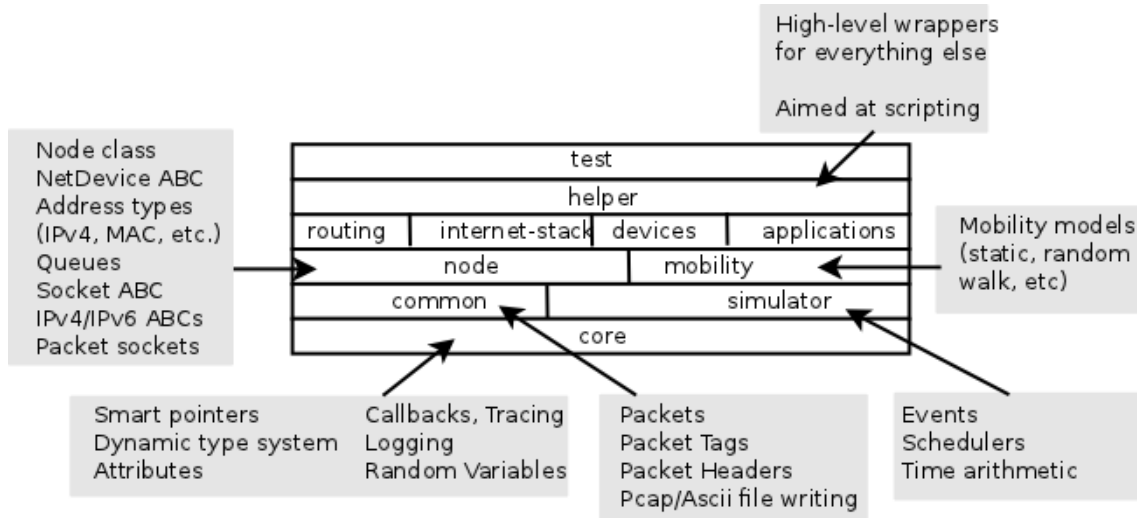


Figura A.1: Módulos existentes en ns-3.

El código fuente de ns-3 se encuentra en su mayoría en el directorio `src`. Como se aprecia en la figura A.1, los módulos solo tienen dependencias de los módulos que están sobre ellos.

El núcleo del simulador (*core*) lo forman los componentes que son comunes en todos los protocolos, hardware y modelos de entorno. El núcleo está implementado en `src/core`. Los paquetes son objetos fundamentales en un simulador de red, y están implementados en `src/network`. Estos dos módulos por sí mismos forman el núcleo de un simulador genérico que puede usarse en diferentes tipos de redes. Los módulos que aparecen por encima de estos dos módulos son independientes de los modelos de red o de los dispositivos de red, y cubren los diferentes módulos del simulador.

En el siguiente apartado se muestran los módulos empleados en el desarrollo de los escenarios de este trabajo.

## A.2 Módulos de ns-3 utilizados

Aunque ns-3 se compone de gran variedad de módulos, en este apartado veremos con más detalle los que han sido utilizados en el desarrollo del trabajo y se consideran básicos. Estos son los módulos que abarcan Wi-Fi y los elementos de red básicos de ns-3.

### Módulo `network`

Este módulo se compone de los elementos básicos de red, como vimos anteriormente, para que puedan proveer de funcionalidades básicas a cualquier topología y simulación. Este módulo contiene los elementos: `Packets`, `NetDevices`, `Node`, etc. Esta sección se centrará en los elementos de los nodos y dispositivos de red.

- **Node:** se trata de la clase fundamental para diseñar los escenarios, ya que es la que crea los elementos que colocaremos en dichos escenarios. A estos elementos luego se les asocian diferentes características, como veremos dentro del módulo Wi-Fi.

### Módulo `Wi-Fi`

Los nodos de ns-3 pueden contener una variedad de diferentes objetos `NetDevice`, y en ambos casos uno de esos objetos/interfaces puede ser Wi-Fi. En esta sección se describe el objeto `Wi-FiNetDevice` y cómo se crean las redes 802.11.

El módulo Wi-Fi para ns-3 incluye diversas características y prestaciones, tales como: diferentes modelos de pérdidas de propagación, diferentes implementaciones de capa física, de estándar, de tipo de AP (con calidad de servicio o sin ella), etc.

Al crear una red Wi-Fi, se necesitan los siguientes elementos:

- **`YansWi-FiChannelHelper`:** Este *Helper* es el encargado de crear un canal Wi-Fi con un modelo de pérdidas de propagación por

defecto. El modelo de pérdidas de propagación es el que hemos analizado en el capítulo 4, `LogDistancePropagationLossModel`.

- **YansWi-FiPhyHelper:** Esta clase crea un objeto que creará instancias de la capa física, pudiendo añadirle un modelo de movilidad (`MobilityModel`).
- **NqosWi-FiMacHelper** y **QosWi-FiMacHelper:** ambas clases sirven para crear instancias de un objeto `ns3::Wi-FiMac`, los cuales son configurados con parámetros típicos como la clase de MAC. La primera de ellas lo configura de tal manera que no existe calidad de servicio, mientras que en la segunda sí. Es por esto que nosotros en nuestro trabajo utilizaremos `QosWi-FiMacHelper`.
- **Wi-FiHelper:** Es la clase que se encarga de crear los `Wi-FiNetDevices` que asociaremos a los nodos. Para crearlos es necesario haber creado tanto la capa física como la capa MAC.

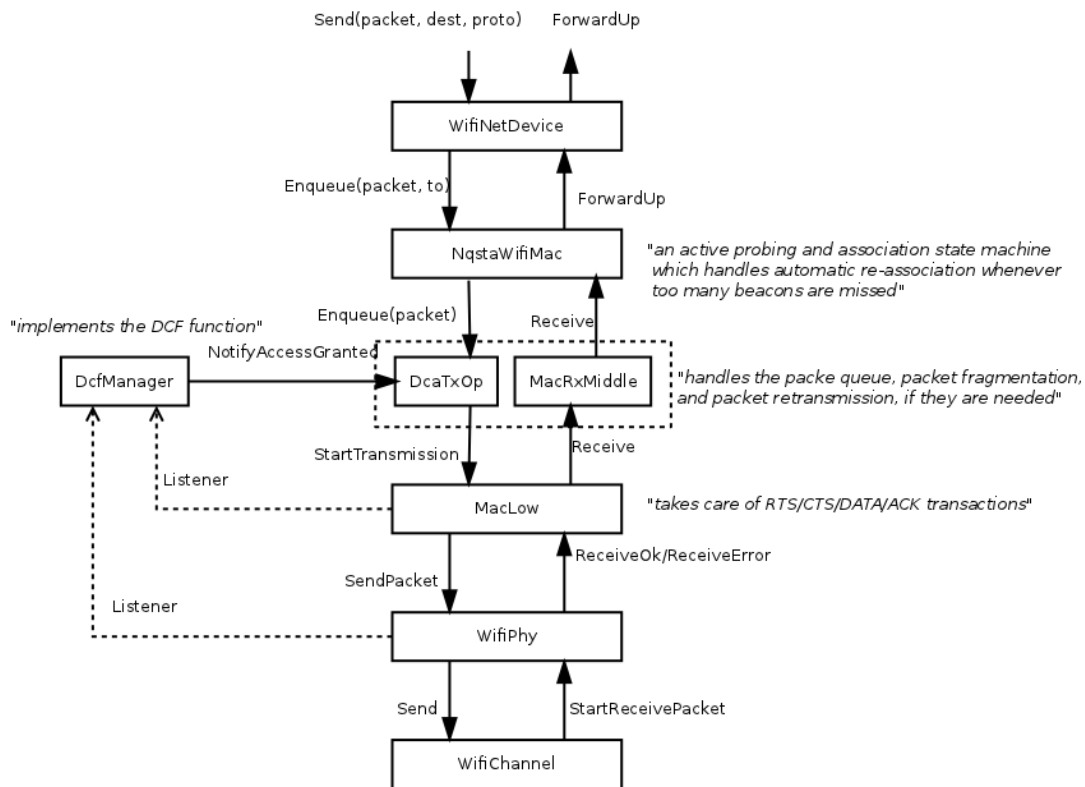


Figura A.2: Arquitectura de un `Wi-FiNetDevice`.