



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

## Balance de tráfico y provisión de QoS en redes IEEE 802.11

Autor

Javier García Fortún

Director

Ángela Hernández Solana

Escuela de Ingeniería y Arquitectura  
2015





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>. Javier García Fortún

con nº de DNI 73015954Q 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) Grado \_\_\_\_\_, (Título del Trabajo)

Balance de tráfico y provisión de QoS en redes IEEE 802.11

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

Zaragoza, 7 de Septiembre de 2015

Fdo: Javier García Fortún



# **Balance de tráfico y provisión de QoS en redes IEEE 802.11**

## **RESUMEN**

El espectro radioeléctrico empleado por los dispositivos que funcionan bajo los estándares IEEE 802.11, englobados bajo la marca comercial Wi-Fi, es un medio compartido con otros estándares como Bluetooth o ZigBee y con otros equipos del mismo sistema de forma que los recursos disponibles son muy limitados y se encuentran saturados. Además, los despliegues de redes Wi-Fi en edificios y entornos públicos y privados para proporcionar un servicio de acceso a internet a clientes y usuarios son cada vez más habituales.

En estos despliegues planificados en modo infraestructura, la gestión de los recursos radio a través de una adecuada planificación de frecuencias, pero también de mecanismos lo más eficientes posibles de ajuste/reajuste de cobertura, selección inteligente de puntos de acceso, algoritmos de balanceo de carga lo más efectivos posibles y de adaptación de tasa en función del estado de la red, se antoja fundamental para conseguir satisfacer las elevadas exigencias que los usuarios presentan en la actualidad. Por otra parte, la demanda de servicios con mayores tasas de descarga y de transmisión para servicios best-effort y de servicios de bajo retardo (VoIP o videojuegos en línea) es cada vez más habitual.

El trabajo fin de grado ha consistido en la implementación en el software de código libre Network Simulator 3 y su posterior evaluación de diferentes soluciones y estrategias de balanceo de tráfico entre distintos puntos de acceso en despliegues multicelulares de redes de área local IEEE 802.11g.

Para evaluar y comparar las distintas estrategias de una forma realista se han considerado distribuciones de usuarios y de tráficos variables a la vez que se han empleado los parámetros establecidos en las normas IEEE 802.11 y se ha modelado las condiciones del canal radio de una forma lo más realista posible.

Como conclusión se han elaborado unos algoritmos que, a partir de los múltiples resultados que se han obtenido de las diferentes simulaciones realizadas, intentan maximizar la calidad de servicio que una red de área local puede ofrecer al usuario, marcando como objetivo maximizar el throughput y reducir el retardo que deben soportar en sus transmisiones.

De forma adicional, se ha estudiado el estado actual de los estándares IEEE 802.11 y de las posibilidades que existen para gestionar los recursos radio y para implementar en sistemas reales las alternativas propuestas en el trabajo.

# Traffic balance and QoS enhancement in IEEE 802.11 networks

## SUMMARY

The radio spectrum used by devices that operate under the IEEE 802.11 standards, which are included below the Wi-Fi trademark, is shared with other standards such as Bluetooth or ZigBee and other 802.11 equipments so the available resources are very limited and they are saturated. In addition, the Wi-Fi networks deployments in public and private environments to provide Internet access service to customers and users are becoming more commonly used.

In these planned deployments in infrastructure mode, the radio resource management through proper frequencies planning, but also the most efficient possible set/reset coverage mechanisms, intelligent selection of access points, most effective load balancing algorithms possible and rate adaptation depending on the network status, seems essential for satisfy the high demands which users have today. Moreover, the demand for higher download and upload rates for best-effort services, but low latency services (VoIP or online gaming) is becoming more popular.

The final project has involved the implementation and subsequent evaluation of different solutions and strategies for balancing traffic between different access points in a multicellular deployment of IEEE 802.11g local area networks in the open source software Network Simulator 3.

To evaluate and compare the different strategies realistically, variable distributions of users and changeable traffic have been considered, as well as we have used the parameters established in the IEEE 802.11 standards and channel conditions are been modeled the most realistic way possible.

As a conclusion, algorithms have been developed which, from the multiple results obtained from the different simulations, try to maximize the quality of service that a local area network can provide to the user. Its target is maximizing the throughput and reducing the delay which users must suffer in their transmission.

In addition, the current state of the IEEE 802.11 standard has been studied and the possibilities to manage radio resources and to implement in real systems solutions proposed in this work.



# Índice de contenidos

1. Introducción.....	1
1.1. Motivación y objetivos.....	1
1.2. Herramientas.....	2
1.3. Estructura de la memoria.....	3
2. Estado del arte.....	6
2.1. Estándares 802.11.....	6
2.2. Protocolos de acceso al medio.....	7
2.2.1.Distributed Coordination Function (DCF).....	7
2.2.2.Point Coordination Function (PCF).....	10
2.3. Descubrimiento y asociación de redes.....	10
2.4. Estrategias para mejorar el rendimiento de redes WLAN 802.11.....	11
2.4.1.Adaptación tasa y potencia.....	11
2.4.2.Balanceo de carga.....	12
2.4.2.1 Algoritmos de selección de AP.....	13
2.4.2.2 Algoritmos de respiración celular o cell breathing.....	14
3. Modelo de sistema.....	16
4. Propuestas.....	22
4.1. Algoritmos de selección de tasa.....	22
4.2. Algoritmos de asociación y roaming.....	23
4.2.1.Algoritmos de optimización del canal radio.....	24
4.2.2.Algoritmos de balanceo del tráfico.....	25
4.2.3.Algoritmos mixtos.....	27
4.2.3.1 Algoritmo mixto 1.....	28
4.2.3.2 Algoritmo mixto 2.....	30

5. Resultados.....	32
5.1. Condiciones de evaluación.....	32
5.2. Umbrales experimentales.....	34
5.2.1.Umbral RSSI/SINR.....	34
5.2.2.Umbral distancia.....	36
5.2.3.Umbral tasa cursada.....	36
5.2.4.Umbral carga del canal.....	38
5.3. Evaluación algoritmos básicos.....	39
5.4. Evaluación algoritmos mixtos.....	41
6. Conclusiones y líneas futuras.....	48
6.1. Conclusiones.....	48
6.2. Líneas futuras.....	49
Anexo A. Acrónimos.....	54
Anexo B. Introducción a Network Simulator 3.....	58
B.1 Introducción.....	58
B.2 Configuración de la simulación.....	59
B.3 Modelo Wi-Fi en ns-3.....	62
Anexo C. Soporte para despliegues 802.11 multicanal sin solape espectral en ns-3.....	69
C.1 Limitaciones de ns-3.....	69
C.2 Implementación en ns-3.....	71
Anexo D. Implementación de algoritmos de asociación en redes en modo infraestructura en ns-3..	77
D.1 Métricas.....	77
D.2 Implementación.....	80
Anexo E. Estándares 802.11.....	89
E.1 Normas 802.11 de nivel físico.....	89

E.2 Otras normas 802.11 importantes.....	90
E.3 Modelo de gestión 802.11.....	91
Anexo F. Estándar IEEE 802.11k.....	95
F.1 Nuevos formatos de las tramas de gestión.....	95
F.2 Mecanismos y medidas de RRM.....	97
Anexo G. Cronología del trabajo.....	102
Bibliografía.....	104

# Índice de figuras

Figura 2.1: Nodo oculto y nodo expuesto.....	8
Figura 2.2: Formato de transmisión con la opción RTS/CTS habilitada.....	9
Figura 2.3: Escaneo pasivo y activo de redes WLAN.....	11
Figura 2.4: Ejemplo de cell breathing.....	14
Figura 3.1: Topología del escenario de simulación.....	17
Figura 3.2: Efectos del shadowing en el radio de cobertura.....	18
Figura 3.3: Velocidad física en función del tamaño del paquete en el nivel de aplicación.....	20
Figura 3.4: Radios de cobertura en función del MCS.....	20
Figura 4.1: Modelo de errores de las modulaciones OFDM (802.11a/g) y DSSS(802.11b/g) en Yans .....	28
Figura 4.2: Relación ocupación del canal y RSSI en el algoritmo mixto 2.....	29
Figura 5.1: Movimiento rectilíneo variable.....	33
Figura 5.2: Throughput global del escenario – Márgenes RSSI/SINR.....	35
Figura 5.3: Retardo de transmisión medio – Márgenes RSSI/SINR.....	35
Figura 5.4: Throughput global y retardo de transmisión medio – Márgenes distancia.....	36
Figura 5.5: Número de traspasos – Márgenes distancia.....	36
Figura 5.6: Throughput global del escenario – Umbrales tasa cursada.....	37
Figura 5.7: Retardo de transmisión medio – Umbrales tasa cursada.....	37
Figura 5.8: Throughput global del escenario – Umbrales carga del canal.....	38
Figura 5.9: Retardo de transmisión medio – Umbrales carga del canal.....	38
Figura 5.10: Throughput global y retardo de transmisión medio – Comparativa criterios – 1Mbps/1024B.....	40
Figura 5.11: Número de traspasos y ocupación del canal – Comparativa criterios – 1Mbps/1024B.	40
Figura 5.12: Throughput global del escenario – Márgenes algoritmo mixto1.....	41
Figura 5.13: Retardo de transmisión medio – Márgenes algoritmo mixto1.....	42

Figura 5.14: Throughput global del escenario – Umbrales algoritmo mixto2.....	42
Figura 5.15: Retardo de transmisión medio – Umbrales algoritmo mixto2.....	43
Figura 5.16: Throughput global y retardo de transmisión medio – Comparativa criterios – 1Mbps/512B.....	43
Figura 5.17: Throughput global y retardo de transmisión medio – Comparativa criterios - 1Mbps/2048B.....	44
Figura 5.18: Throughput global y retardo de transmisión medio – Comparativa criterios – 2Mbps/512B.....	44
Figura 5.19: Throughput global y retardo de transmisión medio – Comparativa criterios – 2Mbps/2048B.....	45
Figura 5.20: Retardo de transmisión y tasa de pérdidas de paquetes – Comparativa criterios – 64kbps/172B.....	46
Figura B.1: Clases de ns-3 del módulo Wi-Fi.....	63
Figura B.2: Máquina de estados de las estaciones móviles en modo infraestructura.....	66
Figura C.1: Solapes entre canales Wi-Fi.....	70
Figura C.2: Modelo de clases con soporte multicanal.....	75
Figura D.1: Máquina de estados propuesta para las estaciones móviles en modo infraestructura.....	87
Figura E.1: Modelo de gestión 802.11.....	93
Figura E.2: Diagrama de secuencias de una solicitud de medidas.....	100
Figura G.1: Diagrama de Gantt.....	102

# Índice de tablas

Tabla 3.1: Modelo de propagación y parámetros de nivel físico.....	18
Tabla 3.2: Parámetros de nivel MAC empleados en 802.11g.....	19
Tabla 4.1: Métricas y sus posibles márgenes/umbrales de reasociación.....	27
Tabla 4.2: Pesos del algoritmo mixto 1.....	29
Tabla 5.1: Parámetros de generación de tráfico.....	32
Tabla 5.2: Parámetros generales de la simulación.....	33
Tabla 5.3: Métricas y sus umbrales de reasociación definitivos.....	39
Tabla B.1: Comparación ns-2 y ns-3.....	58
Tabla C.1: Utilización de la banda de 2,4GHz para Wi-Fi en Europa.....	69
Tabla E.1: Especificaciones de nivel físico de estándares 802.11.....	89
Tabla G.1: Cronología del trabajo.....	102

# Índice de códigos

Código B.1: Ejemplo de configuración de un nodo con interfaces CSMA/CD y CSMA/CA.....	59
Código C.1: Script de configuración del escenario con soporte multicanal.....	69
Código C.2: Configuración de las aplicaciones en el escenario con soporte multicanal.....	70
Código C.2: Gestión de la cola y los sockets en la clase StaWifiMac.....	71
Código D.1: Variables necesarias para los algoritmos de decisión en la clase StaWifiMac.....	78
Código D.2: Variables necesarias para los algoritmos de decisión en la clase StaWifiMac.....	78
Código D.3: Implementación de un algoritmo de decisión simple en la clase StaWifiMac.....	80
Código D.4: Implementación del algoritmo de decisión mixto1 en la clase StaWifiMac.....	82
Código D.5: Implementación del algoritmo de decisión mixto2 en la clase StaWifiMac.....	82



# 1. Introducción

## 1.1. Motivación y objetivos

Durante los últimos años, desde que las primeras normas IEEE 802.11 fueran lanzadas al final del siglo pasado, estos estándares se han convertido en la tecnología dominante de las redes WLAN. Su bajo coste unido al empleo de bandas de uso libre como las ISM (*Industrial, Scientific and Medical*) han hecho desde el primer momento esta tecnología muy atractiva a los fabricantes.

El uso de las bandas ISM, con un ancho de banda muy limitado, junto con el elevado número de despliegues Wi-Fi que podemos encontrar, ha provocado una sobreexplotación de los recursos radio que, sumada a la incipiente necesidad de tráfico requerida por la creciente demanda de servicios en tiempo real, está provocando que las necesidades que demandan los usuarios no puedan ser satisfechas, degradando con ello dramáticamente la experiencia del usuario en servicios como la VoIP o el streaming de vídeo en alta definición.

Al contrario que las redes cableadas, cuyas versiones FFTx y HFC están aumentando rápidamente sus velocidades para soportar las crecientes demandas de los usuarios, las redes inalámbricas IEEE 802.11 no están pudiendo soportar este crecimiento ya que aunque las tasas máximas definidas en las nuevas normas si estén creciendo exponencialmente, hasta los 7Gbps en el último estándar publicado, en la práctica las tasas reales que los usuarios pueden disfrutar son mucho más bajas.

Por otro lado, las redes WAN están acelerando su crecimiento a través de las tecnologías LTE permitiendo a los usuarios disfrutar de conexiones de alta velocidad y bajo retardo desde cualquier punto y con unos niveles de itinerancia superiores a las redes WLAN IEEE 802.11. Afortunadamente para los fabricantes de dispositivos compatibles con IEEE 802.11, las redes 4G todavía no son óptimas para ser conexiones de uso permanente debido a su elevado coste en la mayoría de los países.

Para poder competir con estas nuevas tecnologías y ofrecer a los usuarios la experiencia y calidad que están demandando en sus servicios será necesario realizar una gestión óptima de los limitados recursos radio de los que disponen los equipos Wi-Fi.

Este trabajo se centra en estudiar despliegues planificados de redes inalámbricas basadas en Wi-Fi en modo infraestructura. En concreto, el sistema considerado es una red ESS (*Extended Service Set*), es decir, un despliegue formado por varios puntos de acceso que comparten su identificación SSID y se encuentran vinculados entre sí por un DS (*Distribution System*) que en este caso es una red LAN Ethernet a la que se encuentran todos conectados. El hecho de que la infraestructura esté planificada implica que el emplazamiento de los puntos de acceso, su potencia de transmisión y los canales frecuenciales en los que funciona cada uno se efectúa de tal forma que se cumplen unos objetivos determinados de cobertura y se minimiza la interferencia co-canal y de canal adyacente entre las celdas asociadas a cada punto de acceso.

En este contexto, los objetivos específicos de este trabajo van encaminados a aumentar la provisión de QoS (Calidad de Servicio) y a mejorar la experiencia de los usuarios al usar dicha red inalámbrica, orientando el trabajo hacia la evaluación de estrategias dirigidas a obtener la máxima QoS y un uso efectivo de los recursos del despliegue.

- El primer objetivo es efectuar un estudio de alternativas propuestas en la literatura relacionadas con criterios de selección de puntos de acceso y en la posterior itinerancia o roaming de las estaciones móviles en un despliegue WLAN para optimizar el uso de los recursos radio. El objetivo es evaluar el impacto de distintas estrategias de selección consideradas en la literatura y en particular las ventajas potenciales de estrategias alternativas a esquemas clásicos como la selección del punto de acceso del que se recibe la señal con mayor potencia. Es importante indicar que el criterio específico de selección no es un aspecto estandarizado. Por último, se revisarán posibles alternativas sobre la adaptación de potencia y de tasa propuestas en la literatura y se estudiarán las posibilidades que ofrecen las diferentes normas IEEE 802.11 para la gestión de los recursos radio (RRM).
- En segundo lugar, una vez seleccionadas un conjunto de estrategias compatibles con el estándar, se evaluarán mediante simulación las prestaciones utilizando como una medida de comparación la mejora en el nivel de QoS. En concreto se intentará maximizar el throughput, reducir la latencia y el número de tramas perdidas. Además, también se comprobará en el análisis el impacto que tiene el empleo de unas estrategias en la adaptación de la tasa de transmisión frente a otras.

## 1.2. Herramientas

El proyecto se basa en la evaluación mediante simulación de las distintas propuestas seleccionadas utilizando el software Network Simulator 3, en el que además de implementar y configurar el despliegue Wi-Fi usando modelos ya disponibles, se realizarán todas las modificaciones en el código necesarias que permitan la evaluación de las estrategias seleccionadas. Tal y como se explica a lo largo de la memoria, el modelo suministrado con el simulador presenta algunas limitaciones, como el no soporte de escaneo en múltiples canales de forma simultánea, que deben ser subsanadas incorporando grandes modificaciones que se encuentran descritas detalladamente en los anexos C y D.

Por otra parte, a pesar de que todas las simulaciones finales se han realizado en el software Network Simulator 3, se han empleado otras herramientas que han facilitado la depuración y verificación de los modelos y escenarios empleados en el simulador.

- Network Simulator 3  
Network Simulator 3 o ns-3 es el simulador que se ha empleado en este trabajo. En el anexo B se encuentra una breve explicación de su estructura y del modelo Wi-Fi que tiene implementado. Otras dos herramientas que se han necesitado para emplear correctamente este simulador han sido las dos siguientes:

- C++: Tanto el núcleo como los scripts de configuración del simulador emplean este lenguaje de programación.
- Doxygen: Es un generador de documentación completamente gratis con soporte para los lenguajes de programación más extendidos. Gracias a esta documentación se ha podido comprender la estructura de modelos y clases que el simulador tiene implementada.
- Wireshark  
Software analizador de tráfico y protocolos que junto con la herramienta tcpdump incluida por defecto en la mayoría de sistemas operativos Linux se ha empleado para verificar que tanto el modelo suministrado de Wi-Fi por el simulador como las modificaciones que se han realizado devolvieran unos resultados válidos.
- Microsoft Azure y Amazon Web Services (AWS)  
Para acortar los tiempos de simulación se ha aprovechado el periodo de prueba que ofrecen estas dos herramientas para usar sus servicios de alojamiento para crear varias máquinas virtuales en la nube y así tener de forma simultánea un mayor número de simulaciones.

### 1.3. Estructura de la memoria

El trabajo se encuentra estructurado en seis puntos diferenciados entre sí y que tienen los siguientes contenidos:

- El primer capítulo consiste en la **introducción** del trabajo donde se comenta la motivación y los objetivos que se pretenden conseguir en este trabajo. Además también incluye la estructura de la memoria.
- En el segundo capítulo se proporciona una breve descripción del **estado del arte** del funcionamiento de los protocolos IEEE 802.11 y de posibles alternativas relacionadas con estrategias de balanceo de carga entre puntos de acceso, estrategias de adaptación de potencia y adaptación de tasa.
- El tercer capítulo describe el **modelo de sistema** simulado. Aquí se definen y describen todas las variables que condicionan el escenario de configuración, desde la topología hasta los parámetros del estándar Wi-Fi.
- En el cuarto capítulo se describen todas las posibles **propuestas** de algoritmos para balancear de la forma más óptima el tráfico en una red WLAN 802.11.
- El quinto capítulo son los **resultados** que se han obtenido de simular las anteriores y las condiciones en las que se han realizado. Aquí se evaluará el impacto de distintas condiciones de tráfico y la parametrización de las propuestas realizadas.

- Finalmente, en el sexto capítulo se recopilan las principales **conclusiones** que se han podido extraer de los estudios realizados y se incluye una pequeña reflexión sobre las **líneas futuras** que podría tener este trabajo.

Además, al final se encuentran varios anexos que amplían el contenido de esta memoria exponiendo más detenidamente algunos aspectos que son explicados someramente o simplemente nombrados a lo largo del trabajo.

- En el **anexo A** se encuentran todas las abreviaturas que se han empleado tanto en la memoria como en los anexos.
- El **anexo B** es una introducción al empleo del simulador ns-3 y contiene una breve descripción del mismo y del modelo de clases que simulan las redes y equipos Wi-Fi.
- El **anexo C** describe los cambios que son necesarios en el núcleo del simulador para soportar múltiples canales Wi-Fi dentro de un mismo nodo en simulaciones configuradas en modo infraestructura.
- El **anexo D** complementa al anexo anterior y contiene además las modificaciones que se han realizado en algunas clases del simulador para ampliar las funcionalidades de las estaciones móviles en el modo infraestructura y soportar una gestión inteligente de los trasposos entre diferentes puntos de acceso.
- El **anexo E** contiene una revisión del estado del arte de los estándares 802.11 más importantes.
- En el **anexo F** se encuentra descrito de una forma breve los aspectos más relevantes, y que están relacionados con los objetivos de este trabajo, del estándar IEEE 802.11k, destinado a facilitar la gestión de los recursos radio (RRM).
- El **anexo F** contiene la cronología del trabajo con un desglose de las actividades realizadas y el tiempo invertido en ellas.



## 2. Estado del arte

### 2.1. Estándares 802.11

La familia de estándares 802.11 consiste en una serie de técnicas y especificaciones definidas por el IEEE (Institute of Electrical and Electronics Engineers) para la creación de una arquitectura común del nivel físico y de enlace del modelo OSI que permitan la creación de redes WLAN con transmisiones a través de canales radio. Todos los productos y equipos que cumplen la familia de estándares de 802.11 se engloban bajo la marca comercial Wi-Fi.

Una característica común de los dispositivos Wi-Fi es que funcionan en las bandas de uso sin licencia denominadas ISM por sus siglas en inglés (Industrial, Scientific and Medical). La enorme expansión que las redes Wi-Fi han sufrido en los últimos años se debe a la gran ventaja que supone trabajar en estas bandas ya que no es necesario pagar un canon por usarlas. Sin embargo, esta ventaja también es el mayor inconveniente que presentan estas redes ya que la gran mayoría de equipos del mercado trabajan únicamente en la banda de 2,4GHz, cuyas frecuencias se encuentran fuertemente saturadas al ser insuficiente el ancho de banda que presenta la banda para cubrir la creciente demanda. Además, hay que sumar que esta banda sufre interferencias por parte de otros sistemas como el cableado de luz de alta tensión, los hornos microondas y otros equipos que empleen tecnologías Bluetooth o ZigBee.

Para paliar este problema las últimas normas ya contemplan transmisiones en otras bandas y, aunque la versión 802.11a ya contempló el uso de la banda de 5GHz, apenas se fabricaron dispositivos que trabajaban en esta banda por lo que no ha sido hasta la implantación masiva de equipos compatibles con la norma 802.11n cuando se ha empezado a explotar esa parte del espectro radioeléctrico.

Los estándares se pueden dividir en dos grandes grupos:

- El primer grupo corresponde con las enmiendas, normas o simplemente correcciones que publican los diferentes grupos de trabajo (*Task Group*) que son creados para alcanzar objetivos más concretos como puede ser incluir provisión de QoS, aumentar la seguridad o mejorar la itinerancia.
- El segundo grupo engloba los únicos documentos que el IEEE reconoce como estándares oficiales. Estos incluyen las diferentes normas del grupo anterior que se han ido publicando a lo largo de los años y que actualiza la arquitectura del anterior estándar. La versión más reciente es la 802.11-2012.

En el anexo E se encuentra un resumen de las normas 802.11 más importantes tanto de nivel físico como de las relacionadas con la provisión de QoS, mejora en la gestión de los recursos radio y gestión de la itinerancia.

La norma que más aceptación ha tenido por parte de los fabricantes ha sido la 802.11g [1] que llegó al mercado incluso antes de que se estandarizara. El principal motivo es que fue diseñado desde su origen para ser completamente retrocompatible con los dos estándares anteriores (802.11a/b). Para ello definió cuatro posibles capas físicas, conocidas como *Extended Rate Physical* (ERP), que combinan las modulaciones OFDM (heredadas del 802.11a) con las modulaciones DSSS/CKK definidas en 802.11b. Funciona, al igual que la versión b, sobre la banda de 2,4GHz pudiendo obtener una tasa física máxima de 54Mbps que a nivel de aplicación se convierte en 31,4Mbps. En la actualidad la norma 802.11g está siendo desplazada por la 802.11n [2] que aporta interesantes mejoras que redundan en una mayor velocidad de transmisión. Otra enmienda importante desde el punto de vista de la mejora en la gestión del roaming y de los recursos radio es la 802.11k [3], que introduce y amplía la información y los procedimientos disponibles para mejorar la fiabilidad y la calidad de las redes WLAN (ver anexo F para más información).

## **2.2. Protocolos de acceso al medio**

Puesto que el espectro radioeléctrico es un medio limitado y compartido por todos los dispositivos que funcionan bajo la certificación IEEE 802.11, es necesario un protocolo de nivel MAC para permitir que las estaciones puedan compartir el medio de un modo ordenado y justo. El primer protocolo lanzado oficialmente, el 802.11-1999, contempló inicialmente dos modos de funcionamiento en función de si existe un punto de acceso a través del cual debe pasar todo el tráfico (modo infraestructura) o si las estaciones se pueden comunicar entre sí directamente (modo ad hoc). En ambos modos de funcionamiento el protocolo de nivel MAC está basado en el mecanismo CSMA/CA, donde una estación que quiere transmitir comprueba antes si el medio está ocupado y si determina que está ocupado entonces no transmite.

En el caso del modo infraestructura se definieron dos modos de operación (DCF y PCF) en función de cómo se coordinaban las estaciones para acceder al medio aunque posteriormente estos dos modos fueron expandidos y mejorados en la norma 802.11e para soportar diferenciación de tráfico y añadir herramientas para proveer una cierta QoS.

### ***2.2.1. Distributed Coordination Function (DCF)***

Es el modo de operación más extendido y está orientado a soportar servicios asíncronos ya que en su versión inicial no aplicaba ninguna política de QoS.

Está basado en el mecanismo CSMA/CA, por lo que cuando una estación quiere transmitir comprueba antes si el medio está ocupado. Si la estación encuentra el medio libre transmite pero cuando el medio está ocupado escucha la trama que se está transmitiendo para conocer su duración e iniciar el contador decreciente denominado NAV (*Network Allocation Vector*) que indicará a la estación cuanto tiempo debe retrasar como mínimo su transmisión para evitar una colisión. El mecanismo CSMA/CA introduce una serie de intervalos de espera que tienen como objetivo la

priorizaron del envío de tramas de gestión o de control de errores. Además, para aleatorizar el acceso al medio una vez que este se encuentre libre y evitar que todas las estaciones intenten acceder al medio al mismo tiempo se establece una ventana de contienda (CW) que establece el tiempo de *back-off*, tiempo adicional que cada nodo debe esperar para acceder al medio y cuyo tamaño va decreciendo con cada intento de acceso fallido.

Otro elemento que se introdujo para resolver el problema de colisiones derivadas de los problemas denominados terminal oculto y terminal expuesto [4][5] (ver figura 2.1) fue el mecanismo RTS/CTS (*Ready to Send/Clear to Send*).

- El problema del nodo oculto se produce cuando el nodo A está transmitiendo hacia el nodo C y el nodo B, que no está en el radio de cobertura del nodo A, detecta el canal libre y comienza a transmitir hacia el nodo C produciéndose una colisión y fallando ambas transmisiones.
- El problema del nodo expuesto se produce cuando una estación (nodo B) no transmite hacia el nodo C a pesar de poder hacerlo porque detecta que el nodo A ya está realizando una transmisión. En este caso, aunque el nodo B esté en la zona de cobertura del nodo A, el nodo C no lo está, por lo que se ha abortado una transmisión que podría haberse realizado sin sufrir ninguna colisión en el equipo receptor.

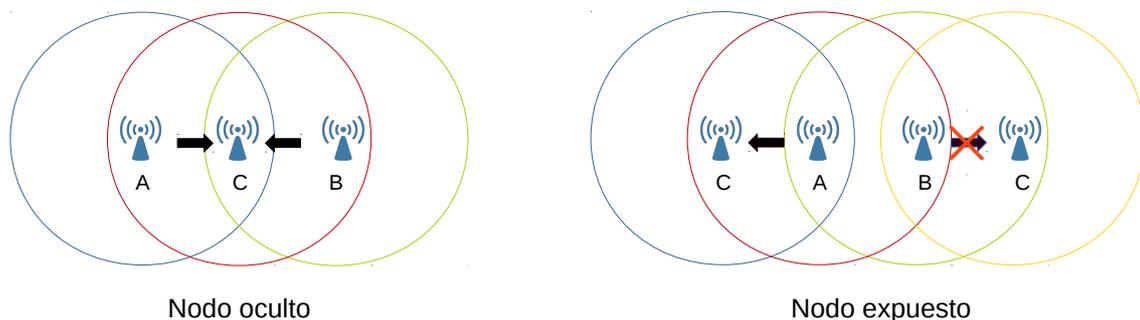


Figura 2.1: Nodo oculto y nodo expuesto

El mecanismo RTS/CTS consiste en el intercambio de un par de tramas entre las dos estaciones implicadas en la comunicación antes de iniciar la transmisión de los datos. La monitorización de dichas tramas por parte de otras estaciones permite identificar el inicio de una transmisión de un terminal oculto. Dado que el intercambio previo de estas tramas supone un incremento de overhead, su habilitación es opcional y solo tiene sentido usarlo en escenarios con un nivel de carga notable o en tramas de datos de gran tamaño.

Para optimizar al máximo el mecanismo CSMA/CA se definieron inicialmente tres intervalos temporales que determinarían el comportamiento de las estaciones cuando quisieran acceder al medio, responder a una transmisión o recuperarse de un error. Debe tenerse en cuenta que todas las transmisiones requieren el envío de paquetes de reconocimiento (ACK) por parte de las estaciones receptoras.

- **DIFS (Distributed Interframe Space):** Tiempo que deben esperar como mínimo las estaciones para conseguir el acceso al medio tanto cuando van a transmitir un RTS como una trama de datos.
- **SIFS (Short Interframe Space):** Intervalo temporal de espera mínimo, y por tanto el que más prioridad tiene, empleado para dos tramas que pertenecen a la misma transmisión, es decir, la respuesta CTS a un RTS o el ACK a una trama de datos. El valor del SIFS está determinado por la capa física, ya que depende de la tasa máxima de transmisión, del tiempo que le cueste al emisor modificar el estado de su receptor al modo de escucha y del tiempo de procesado MAC de la trama.
- **EIFS (Extended Interframe Space):** Es el tiempo de espera empleado por una estación cuando ésta recibe una trama que no ha podido decodificar correctamente. Se calcula en función del tiempo que cuesta transmitir un ACK a la mínima tasa disponible y su función es priorizar la comunicación que está en curso, evitando posibles colisiones.

En la figura 2.2 se puede ver el esquema típico de transmisión determinado por CSMA/CA con el mecanismo RTS/CTS activado.

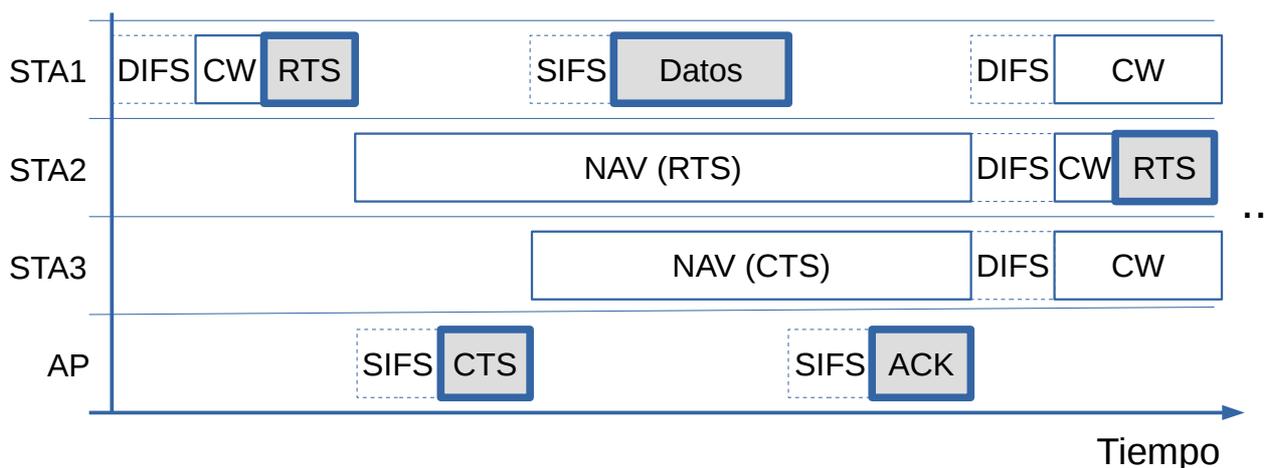


Figura 2.2: Formato de transmisión con la opción RTS/CTS habilitada.

Otros intervalos temporales de control no contemplados inicialmente pero que fueron añadidos en posteriores revisiones como la 802.11e o 802.11n son los siguientes:

- **AIFS (Arbitration Interframe Space):** Intervalo temporal introducido en 802.11e para priorizar entre sí las cuatro categorías de acceso definidas en la norma. El valor del AIFS se configura en las STAs (estaciones) a los valores por defecto para las diferentes categorías si no es transmitido por el AP (punto de acceso) en las tramas Beacon y Probe Response.
- **RIFS (Reduced Interframe Space):** Espacio introducido en la norma 802.11n para aumentar la eficiencia de las transmisiones. El tiempo RIFS es el que una misma estación

debe esperar para realizar múltiples transmisiones y es usado cuando no se espera la recepción de ACKs para cada trama transmitida.

### 2.2.2. Point Coordination Function (PCF)

Es el segundo modo de operación y está destinado a soportar servicios síncronos en despliegues en modo infraestructura con ciertos requisitos de calidad de servicio que el AP puede asegurar a una estación ya que actúa como coordinador asignando espacios temporales a las STAs mediante *polling*. A los intervalos temporales anteriores se le suma otro más llamado PIFS:

- **PIFS (PCF Interframe Space):** Tiempo que esperan los APs para conseguir el acceso al medio. Es un tiempo de slot inferior que el DIFS para otorgar una mayor prioridad a los APs.

## 2.3. Descubrimiento y asociación de redes

El estándar establece un conjunto de tramas de gestión mediante las cuales las estaciones pueden anunciar su presencia y asociarse entre ellas. El proceso de descubrimiento y asociación implica, particularizado en este caso para el modo infraestructura, el intercambio de las siguientes tramas de gestión. El descubrimiento de redes puede ser tanto de una forma pasiva como activa.

- **Beacon:** Su monitorización se asocia a un proceso pasivo de búsqueda. Estas tramas son enviadas periódicamente por los puntos de acceso para anunciar su presencia e informar sobre su identificación (SSID), su hora (timestamp) y otros parámetros como las posibles tasas de transmisión que soporta. Como se verá más adelante, algunas normas 802.11 permiten incluir mayor información en estas tramas para facilitar la RRM.

Las STAs monitorizan durante un cierto tiempo cada canal frecuencial para detectar los Beacon enviados por los APs (en modo infraestructura) o de otras STAs (en modo ad hoc) para elaborar una lista de las redes encontradas. Además, el nivel de potencia de estas tramas es un criterio que se aplica de manera bastante generalizada para seleccionar un punto de acceso frente a otro.

- **Probe Request/Response:** Par de mensajes que son intercambiados únicamente si la STA tiene configurado el escaneo activo de redes. Cuando la estación envía una trama Request, todos los APs que la escuchan le responden con un mensaje Response para indicarle su presencia y los distintos parámetros del AP (su contenido es muy semejante al de una trama Beacon).

- **Association Request/Response:** Par de tramas que son intercambiadas entre la STA y el AP cuando el primero ha decidido asociarse con el segundo. El par de mensajes puede repetirse varias veces si el AP no está de acuerdo con los parámetros que la STA ha indicado en su mensaje Request.

En la figura 2.3 se puede observar el intercambio de los anteriores mensajes tanto en el caso de que la STA tenga configurado la búsqueda de una forma pasiva (espera la recepción de las tramas Beacon para asociarse) o de una forma activa (solicita información a los puntos de acceso que se encuentren en su radio de cobertura mediante un mensaje Probe Request).

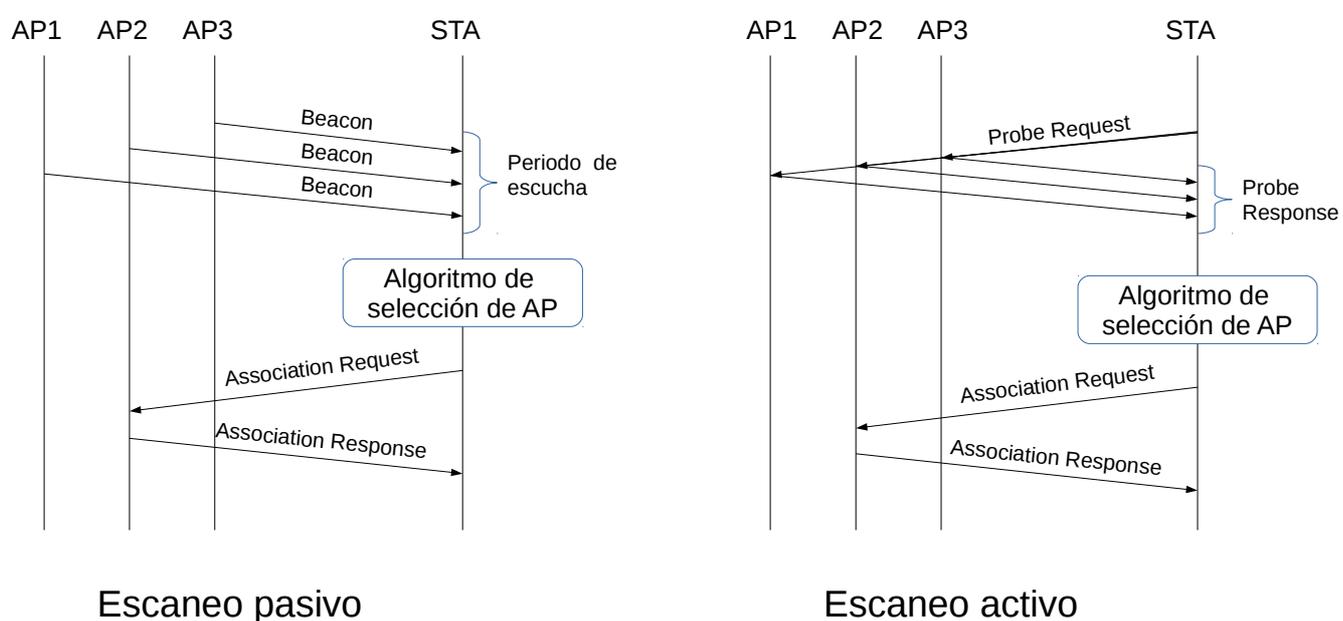


Figura 2.3: Escaneo pasivo y activo de redes WLAN

## 2.4. Estrategias para mejorar el rendimiento de redes WLAN 802.11

### 2.4.1. Adaptación tasa y potencia

Un mecanismo fundamental en el rendimiento global de un despliegue y que no se encuentra estandarizado por el IEEE es el algoritmo de adaptación de la tasa física y de la potencia de transmisión que emplean tanto los puntos de acceso como las estaciones móviles.

Las tasas publicadas en los estándares son las velocidades físicas que se pueden obtener en condiciones ideales donde hay una única estación y que se encuentra lo suficientemente cerca del AP. Aun así, debido al overhead introducido por el protocolo CSMA (tiempos de espera) y las cabeceras tanto del nivel físico como del nivel del enlace, las tasas que el usuario puede disfrutar en el nivel del enlace son netamente inferiores. Por ejemplo, en caso de 802.11b al usar TCP la tasa máxima que podemos disfrutar es de 5,9 Mbps, un 46% de pérdidas respecto al máximo físico.

Además de las pérdidas asociadas al protocolo de acceso, hay que tener en cuenta que cada tasa de transmisión implica unos requisitos de SINR específicos en recepción. La no satisfacción de dichos requisitos conlleva una elevada tasa de error con las consiguientes retransmisiones que implican una fuerte reducción de la eficiencia. Por tanto, elegir la mejor tasa posible considerando las condiciones del canal mediante algoritmos de adaptación de tasa lo más efectivos posibles se antoja fundamental para maximizar el rendimiento de las WLAN.

Existen diferentes algoritmos de adaptación de tasa que se pueden clasificar en dos grupos según de donde toman los datos para tomar las decisiones: El primer grupo está formado por el algoritmo ARF [6] y sus derivados como el AARF [7], AARF-CD [8], CARA [9] que deciden aumentar o disminuir la tasa en función del número de transmisiones satisfactorias o fallidas. El otro grupo está formado por los algoritmos como el RBAR [10] que emplean el nivel de potencia (RSSI) de la anterior trama o CTS recibido para determinar la máxima tasa posible. Tanto en el anexo B como en el apartado 4.2 se comentan y explican varios de estos algoritmos.

En cuanto a los de adaptación de potencia, destinados a conseguir el máximo throughput posible, están muy relacionados con los algoritmos de balanceo de carga mediante el método de respiración celular o “*cell breathing*” que se van a comentar en el siguiente punto.

## 2.4.2. Balanceo de carga

Una de las características que poseen las redes WLAN 802.11 es la libertad que presentan las estaciones para asociarse al punto de acceso que consideren oportuno. Generalmente, no existen políticas de selección de AP o traspaso especificadas en el estándar, siendo la implementación más habitual que las STAs que están bajo la cobertura de dos o más APs, se asocian del que reciben una mayor potencia (RSSI) en las tramas Beacon. Al aplicar como criterio el RSSI se podría asumir a priori que la elección del AP se está realizando para obtener las mejores condiciones de canal y en consecuencia una mayor tasa de transmisión. Sin embargo, debido a la distribución heterogénea de tráfico dentro de las redes Wi-Fi, la aplicación de este criterio puede provocar una excesiva concentración del tráfico en algunos APs de la red, derivando en situaciones de congestión y degradación de las prestaciones.

Cualquier estrategia dirigida a distribuir la carga entre los diversos puntos de acceso de un ESS puede redundar en una mejora global del QoS y del rendimiento global de la red. Así, las estrategias para balancear la carga y optimizar el uso de los recursos radio se pueden distribuir en dos grupos según la forma en la que intentan redistribuir los usuarios en las diferentes celdas. El primer grupo aplica distintas políticas basadas en métricas o variables para decidir cuál es el AP que

debe seleccionarse para mejorar la experiencia tanto del propio STA como del resto. El segundo grupo corresponde a los algoritmos que dan lugar al fenómeno conocido como *cell breathing*, que consiste en aumentar o disminuir la cobertura de los APs para potenciar el enganche de las STAs a un punto de acceso o a otro en función de las necesidades que presente el sistema. Ambas alternativas requieren que el despliegue cuente con amplias áreas de solape entre las diferentes zonas de cobertura.

### 2.4.2.1 Algoritmos de selección de AP

Los criterios de selección de APs tienen, sin lugar a dudas, impacto directo en la QoS. Como se ha comentado, el criterio de selección basado en el RSSI es el más habitual, pero en la literatura se han propuesto otros criterios alternativos ya que la potencia de señal no siempre proporciona la suficiente información sobre las condiciones en el uplink (enlace ascendente) y downlink (enlace descendente). Por ejemplo, un criterio es el nivel de SINR, que es más indicativo de la tasa instantánea que se puede conseguir ya que tiene en cuenta el nivel de interferencia, o incluso otros bastante simples como la elección del AP que se encuentra más cercano.

En cuanto a los algoritmos que emplean métricas que se consideran indicativas del posible desequilibrio o equilibrio existente entre las distintas celdas del despliegue, se pueden encontrar muchos trabajos en la bibliografía. Por ejemplo, en [11] se sugiere el uso de información de estado de los APs, como el número de STAs asociadas y la cantidad de tráfico procesada en el punto de acceso. En [12] se añade un nuevo criterio de selección, el nivel de ocupación del canal, es decir, la fracción de tiempo que el AP detecta que el canal está ocupado durante una ventana temporal determinada. Como se comentará más adelante, las tres variables pueden ser insertadas en campos de información de la trama Beacon. Además, existen otras propuestas [13] que sugieren la elección del AP que es capaz de maximizar una función que pondera un parámetro dependiente de la calidad del canal (por ejemplo, el RSSI) y otra inversamente proporcional al nivel de ocupación del canal.

Algunas soluciones se realizan de una forma descentralizada, puesto que son las propias STAs sin asistencia de ningún AP las encargadas de obtener las métricas necesarias para realizar las estrategias de balanceo de la carga. Las propuestas de la bibliografía buscan variables en el nivel físico o en el nivel MAC que permitan tener una medida representativa del nivel de equidad en la carga de los distintos APs. Algunos ejemplos son los dos siguientes:

- En [14] se proponen dos algoritmos que, a partir del retardo en la respuesta del AP a uno o varios mensajes Probe, las estaciones realizan una estimación de la carga del canal para seleccionar el punto de acceso que presente una menor carga.
- En [15] se especifica la forma en la que las STAs calculan el nivel de SINR en una red 802.11 configurada en el modo de operación PCF. En este caso la STA solo puede calcular de forma autónoma la SINR del downlink. A partir de ella, la STA ya puede determinar la máxima tasa instantánea alcanzable, que es la métrica que se propone emplear para la selección del punto de acceso.

### 2.4.2.2 Algoritmos de respiración celular o *cell breathing*

El *cell breathing* es un mecanismo mediante el cual las celdas con mayor carga reducen su potencia de transmisión de las tramas Beacon con lo que reducen su alcance y en consecuencia su radio de cobertura. Al mismo tiempo, las celdas adyacentes aumentan su potencia para alcanzar las zonas que ha dejado de cubrir la celda sobrecargada.

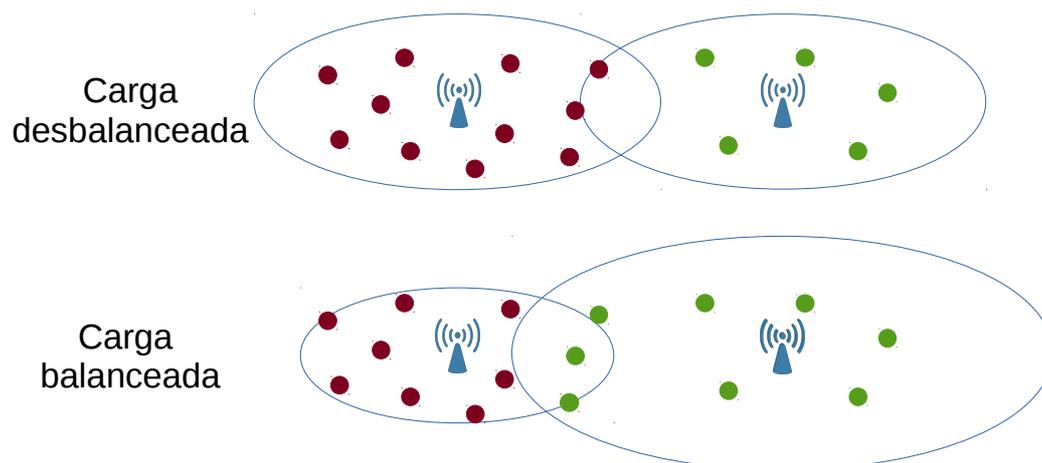


Figura 2.4: Ejemplo de cell breathing

Estas soluciones de naturaleza centralizada implican una topología de red en la que todos los AP de una misma ESS son capaces de comunicarse entre sí para intercambiarse información sobre su estado (nº de STAs, carga del canal...) y actuar para solucionar el desequilibrio. Estas soluciones también consideran el uso de un equipo ajeno al despliegue donde se realizan todos los algoritmos de balanceo de carga.

- En [16] se emplea un equipo central que recibe todos los datos de los APs, ejecuta el algoritmo y comunica a los APs la decisión que ha tomado. Introduce el concepto “fuerza de repulsión”, parámetro dependiente de la distancia y definido como el interés que tiene el AP en seguir dando servicio a cada STA, y que es la medida que se emplea para aumentar o disminuir la potencia de transmisión de los APs.
- En [17] se describe un algoritmo centralizado que trabaja con las características definidas en el estándar 802.11e y cuyo objetivo es balancear el tráfico de sesiones en tiempo real. El algoritmo emplea unos umbrales de paquetes descargados diferentes para audio y vídeo que cuando son superados se realiza un ajuste en las áreas de cobertura.
- En [18] el algoritmo que se encarga de recalcular las potencias de transmisión de cada punto de acceso emplea la métrica AAC (*Available Admission Capacity*) definida en el estándar 802.11k y que representa el tiempo, determinado por el control de admisión, que hay disponible en el canal.



## 3. Modelo de sistema

La topología que se va a simular se encuentra representada en la figura 3.1. En concreto consiste en una red ESS (*Extended Service Set*), es decir, un despliegue formado por varios puntos de acceso que comparten su identificador SSID y vinculados entre sí por un DS (*Distribution System*) que en este caso es una red LAN Ethernet a la que se encuentran todos conectados. La idea de emplear un ESS es que los roaming sean transparentes a partir del nivel LLC (*Logical Link Control*) y no sea necesario realizar reconexiones en el nivel de aplicación debido al cambio de IP que supondría el handover entre dos BSS no pertenecientes al mismo ESS.

El estándar Wi-Fi escogido ha sido el IEEE 802.11g frente a otras versiones, como por ejemplo la 802.11n que es más nueva y ya está ampliamente extendida en los dispositivos actuales, debido a varios motivos:

- El primero es que la implementación de este estándar es más antigua y ha sido empleada en más trabajos por lo que se puede confiar, sin necesidad de realizar una verificación exhaustiva del modelo, en que los resultados que se obtengan de la simulación son realistas.
- La elección de 802.11g frente a 802.11b se debió a que la implementación del primero en los dispositivos actuales es prácticamente del 100% por lo que no habría tenido sentido emplear un estándar tan antiguo y que no se emplea en despliegues sin estar acompañado de al menos de la versión 802.11g.
- Por último se optó por desechar la opción de 802.11n por dos motivos: el primero es que el objetivo de este trabajo de estudiar diferentes algoritmos para el balanceo de la carga no debería verse afectado por el hecho de trabajar con unas tasas superiores. El segundo, y más importante, es que el uso de 802.11n habría implicado el uso de tasas más altas por lo que la duración de las simulaciones habrían reducido drásticamente el número de simulaciones que se habrían realizado.

La topología del sistema que se va a evaluar en este trabajo se encuentra representada en la figura 3.1. En este caso concreto, debido a que la inserción de cada AP con las correspondientes STAs en su zona de cobertura aumenta exponencialmente el tiempo de simulación, se ha optado por acotar el tamaño del despliegue a 9 APs. En cuanto a los canales frecuenciales que emplean cada uno de ellos también se puede ver en la figura 3.1 y el motivo de su elección ha estado condicionado por el simulador ns-3 y se puede ver explicado de forma detallada en el anexo C, pero en resumen, se ha optado por usar canales sin ningún solape espectral por lo que las interferencias entre canales no se han podido estudiar en este trabajo y solo habrá que tener en cuenta la interferencia cocanal.

Las dimensiones del escenario, la separación entre los AP y en consecuencia las áreas de solape de cobertura no se han escogido al azar sino que tras estudiar el balance de potencia de la

cobertura mínima de las tramas Beacon y por lo tanto la tasa mínima obligatoria de transmisión de 1Mbps, se obtuvo que el radio de cobertura era de 120 metros.

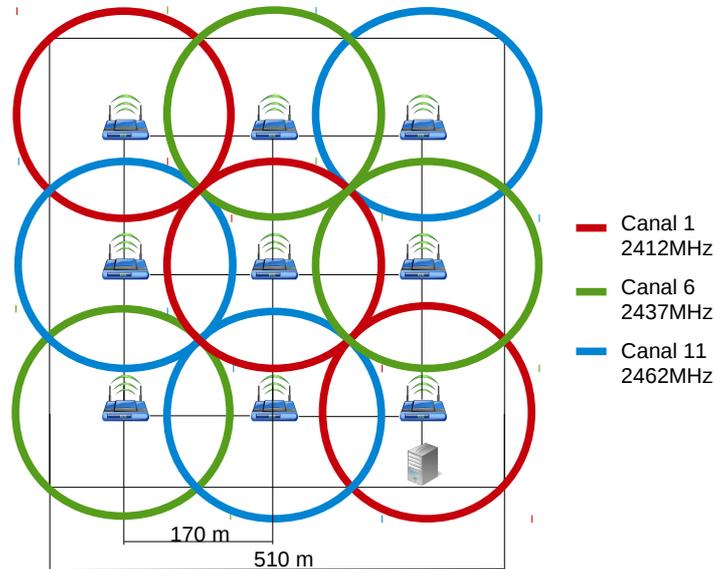


Figura 3.1: Topología del escenario de simulación

Como ya se ha comentado anteriormente, el simulador presenta serias limitaciones en el modelo Wi-Fi relacionadas con el funcionamiento de las STAs en el modo infraestructura. La principal es la incompatibilidad para escanear varios canales de forma simultánea, aunque hay otras como la imposibilidad de realizar un traspaso a otro AP. Para subsanar estos problemas ha sido necesario realizar grandes modificaciones del código del simulador (ver anexos C y D).

En la tabla 3.1 se pueden ver tanto los parámetros del modelo de propagación logarítmico que se ha empleado y que ya se encuentra implementado en ns-3 como también el valor del margen por pérdidas causadas por shadowing. Estas últimas pérdidas se han modelado con una variable aleatoria con una distribución lognormal que se ha implementado para simular el comportamiento del slow fading o shadowing en entornos interiores, es decir, una variación lenta de la potencia de la señal recibida debido a la presencia de obstáculos en el trayecto de propagación. Sus efectos en el radio de cobertura se pueden observar en la figura 3.2, pudiendo darse la circunstancia de que nodos que aparentemente están a una distancia suficiente del AP no reciban suficiente potencia y al contrario, que estaciones fuera del radio de cobertura sufran un nivel de interferencia más elevado del que inicialmente se podría esperar.

$$L = L_0 + 10 \cdot k \cdot \log_{10}\left(\frac{d}{d_0}\right) \quad (3.1)$$

$$L_{total} = L + X_s \quad (3.2)$$

Las ecuaciones 3.1 y 3.2 se corresponden con el modelo de propagación donde  $d$  es la distancia a la que se encuentra la estación receptora desde el punto de transmisión y  $P_t$  y  $P_r$  las potencias de transmisión y recepción respectivamente. En cuanto a las variables  $d_0$ ,  $k$  y  $X_s$ , son parámetros del modelo de propagación cuyos valores se han escogido a partir de algunos valores de referencia obtenidos de forma experimental que se han encontrado a lo largo de la literatura, como por ejemplo en [19][20][21].

Pérdidas referencia $L_0$	46,67	dB	Potencia Tx	18	dBm
Distancia referencia $d_0$	1	m	Ganancia Tx	0	dB
Exponente $k$	3,6		Ganancia Rx	0	dB
Distancia de correlación del shadowing	10	m	Figura de ruido	7	dB
Desviación estándar del shadowing	5	dB	Nº Antenas	1	
Margen fading $X_s$	5	dB	Umbral detección	-96	dBm
			Umbral CCA	-99	dBm

Tabla 3.1: Modelo de propagación y parámetros de nivel físico

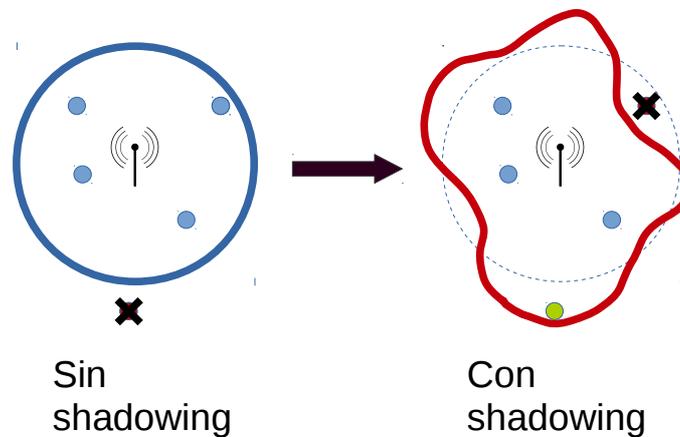


Figura 3.2: Efectos del shadowing en el radio de cobertura

En cuanto a los parámetros del nivel físico (ver tabla 3.1), que corresponderían a las especificaciones de la tarjeta de red de los nodos, se han empleado los valores típicos comerciales como podrían ser los que encontramos en [22]. Para la potencia de transmisión y la ganancia de transmisión también se ha tenido en cuenta el umbral máximo establecido por ley para el PIRE (Potencia Isotópica Radiada Equivalente) que en España [23] está limitado a 20 dB. En este caso al usar una antena omnidireccional alimentada con 18dBm no se supera el PIRE máximo. El número de antenas se ha configurado a uno ya que el estándar 802.11g no contemplaba el uso de antenas MIMO.

En los parámetros de nivel físico también se incluían los umbrales de detección. El primer umbral de detección corresponde con el mínimo nivel de energía que es capaz de demodular correctamente el receptor. El segundo, el umbral CCA (*Clear Channel Assessment*), corresponde con el nivel de potencia a partir del cual el receptor interpreta que el canal está ocupado.

Los parámetros del nivel MAC del estándar 802.11g están representados en la tabla 3.2. Es importante decir que el simulador ns-3 no soporta la coexistencia del estándar 802.11g con el tiempo de slot corto junto con el estándar 802.11b, por lo que la configuración se ha realizado con el tiempo de slot y preámbulo largo y en consecuencia con los tiempos DIFS, PIFS y EIFS también largos. De esta forma, deberemos tener en cuenta que las prestaciones máximas, que serán estudiadas después, serán inferiores a las definidas en el estándar 802.11g [1].

SIFS	10	$\mu\text{s}$
Time Slot	20	$\mu\text{s}$
DIFS	50	$\mu\text{s}$
PIFS	30	$\mu\text{s}$
EIFS	364	$\mu\text{s}$
CTS Timeout	674,67	$\mu\text{s}$
ACK Timeout	674,67	$\mu\text{s}$

Tabla 3.2: Parámetros de nivel MAC empleados en 802.11g

Con las consideraciones anteriores, los primeros cinco parámetros corresponden a los definidos en el modo retrocompatible del estándar 802.11g. En cuanto a los dos últimos timeout, estos valores han sido calculados a partir del tiempo que cuesta transmitir contando su correspondiente SIFS y la propagación por el aire máxima de un ACK/CTS con la mínima tasa de transmisión posible, en este caso 1Mbps.

El nivel MAC tanto de las STA como de las AP se ha implementado sin ningún tipo de QoS por lo que no habrá diferenciación de ningún tipo entre las diferentes STAs. Respecto al tamaño de las colas y el tiempo que debe transcurrir para que se descarte un paquete de ellas, se ha optado por establecer un tiempo elevado (10 segundos) y un tamaño lo suficientemente grande como para que no se descarten paquetes por desbordamiento del buffer.

Otro factor importante a configurar y que tiene un gran impacto en los resultados de la simulación es el uso del mecanismo RTS/CTS. En la gráfica 3.3 se puede observar el resultado que arrojó un conjunto de simulaciones donde se puede ver la tasa máxima que se podría obtener con los parámetros de simulación empleados en condiciones óptimas (una única STA junto al AP).

Como se puede observar, el hecho de no emplear un tiempo de slot corto está limitando el throughput global máximo que se puede conseguir. Además, también se puede observar la fuerte dependencia que presenta respecto al tamaño del paquete ya que el hecho de emplear un paquete de 2048 bytes de nivel de aplicación respecto a uno de 64 bytes con la misma tasa de generación de

tráfico supone un aumento del throughput de casi el 950%. En el caso de usar el mecanismo de protección RTS/CTS, las pérdidas respecto a no usarlo siempre son superiores al 50%, siendo en términos relativos mayor en caso de usar paquetes más pequeños.

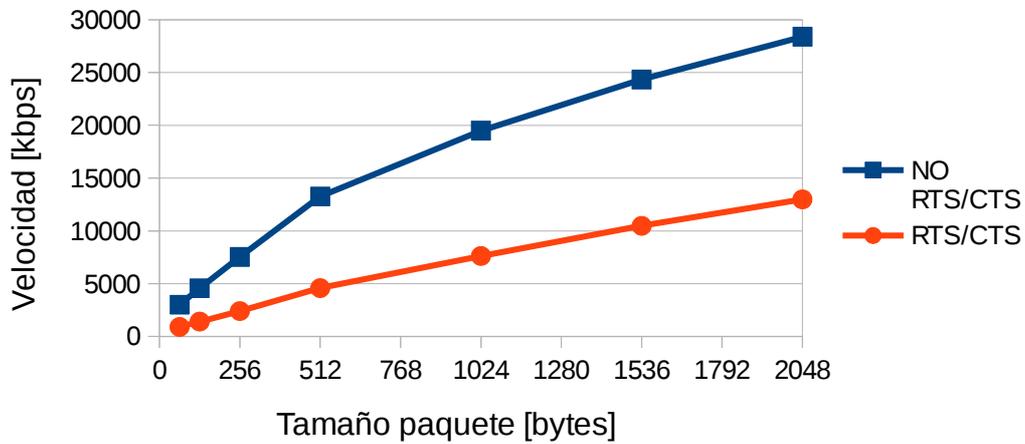


Figura 3.3: Velocidad física en función del tamaño del paquete en el nivel de aplicación

Con el canal, el nivel físico y el nivel MAC ya configurado, ya se puede caracterizar el escenario para conocer algunos parámetros como la potencia media y en definitiva la tasa máxima media a la que las estaciones podrán transmitir. Estas variables dependen de la sensibilidad establecida por el modelo de canal y de errores implementado en el simulador [24]. En la figura 3.4 está representada la relación que existiría entre la distancia y la tasa física en ausencia de interferencias en caso de que las pérdidas por shadowing fueran constantes e iguales al margen establecido de 5dB.

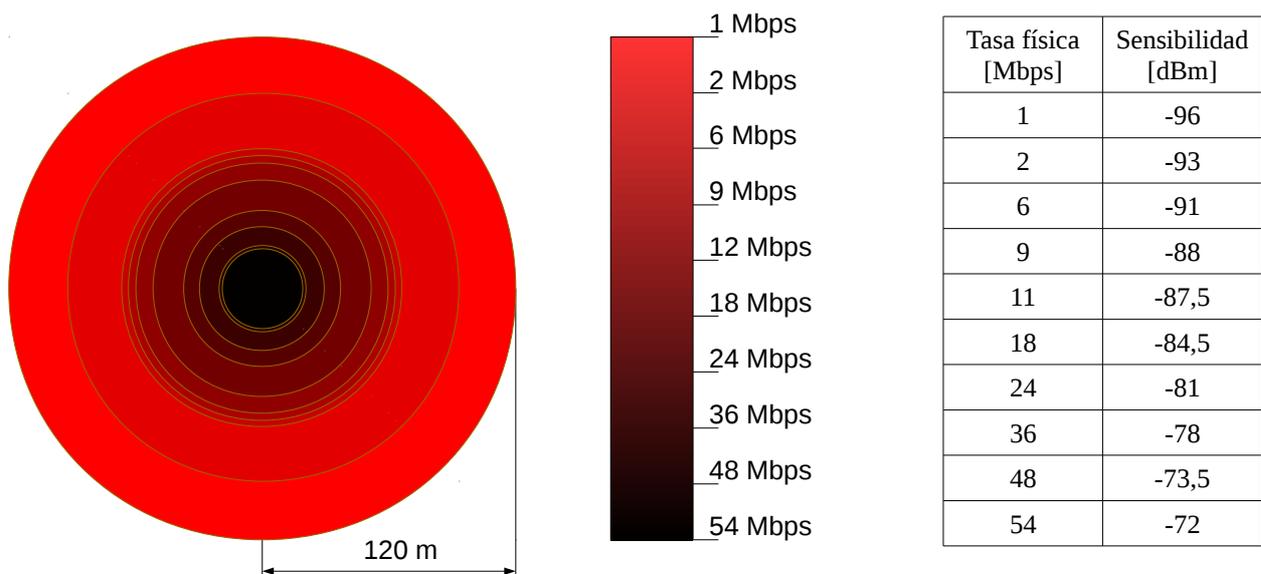


Figura 3.4: Radios de cobertura en función del MCS



## 4. Propuestas

### 4.1. Algoritmos de selección de tasa

Puesto que el algoritmo de selección de tasa tampoco es un elemento que se encuentre estandarizado y su efecto en el rendimiento de una WLAN 802.11 es considerable, se van a probar varios algoritmos para ver el impacto que tienen en la distribución y cantidad de usuarios y del tipo de tráfico que se está generando.

1. AARF (Adaptive Auto Rate Fallback) [7]
  - Es posiblemente el algoritmo más implementado en la mayoría de equipos domésticos y consiste en una versión mejorada del algoritmo ARF [6], introduciendo soluciones a las ineficiencias de este al aumentar de tasa y producirse un error en la transmisión.
  - Su funcionamiento es muy sencillo: Se comienza con la tasa mínima posible (1Mbps en el caso de 802.11g) y cada 10 transmisiones correctas se aumenta la tasa. En caso de que falle después de haber aumentado se vuelve a la tasa anterior y deberá de haber 20 transmisiones satisfactorias para intentar aumentar de nuevo.
  - La gran desventaja de este algoritmo es que fue diseñado para que una estación encontrara la mayor tasa posible en un escenario donde no hay colisiones, por lo que en escenarios multiusuario su rendimiento decae muy rápidamente.
  - Su comportamiento varía en función del umbral del RTS/CTS establecido, así que se va a estudiar con el mecanismo únicamente apagado ya que las dos siguientes versiones optimizan su funcionamiento.
2. AARF-CD (Adaptive Auto Rate Fallback with Collision Detection)[8]
  - Versión mejorada del anterior algoritmo que presenta un mejor comportamiento en escenarios con colisiones ya que es capaz de detectar si una transmisión errónea se ha producido por errores en el canal o por una colisión.
  - El algoritmo no realiza un uso corriente del mecanismo RTS/CTS (generalmente se usa cuando el tamaño de la trama supera un determinado tamaño) sino que lo emplea para determinar si el error en la transmisión anterior se ha producido por un MCS inadecuado o por una colisión.
  - El funcionamiento es el siguiente: cuando el emisor no recibe el ACK correspondiente a la trama de datos que acaba de transmitir envía un mensaje RTS con el mismo esquema de modulación. Si se recibe la respuesta CTS se comprueba que el error en la transmisión se ha debido a una colisión y si no se recibe se supone que el MCS es inadecuado y se baja a uno menos exigente. De esta forma, el algoritmo aprovecha que la colisión de un mensaje corto como el RTS es muy improbable para determinar el origen del error.
  - Para determinar cuándo es necesario enviar un mensaje RTS el algoritmo cuenta con varias ventanas que se van ajustando con el número de transmisiones satisfactorias y erróneas.

### 3. CARA (Collision-Aware Rate Adaptation) [9]

- Algoritmo que también está basado en AARF y que al igual que la versión AARF-CD también emplea el mecanismo RTS/CTS para determinar si las transmisiones fallidas se deben a una colisión o a un MCS incorrecto.
- Además, añade un mecanismo de detección de colisiones empleando el CCA en caso de que el mecanismo RTS/CTS no se haya usado, escuchando al final de la transmisión después de un periodo SIFS. Si no se ha recibido un ACK y el canal está ocupado se asume que ha habido una colisión.
- En este caso el comportamiento del mecanismo RTS/CTS es un poco más complejo que en los dos casos anteriores ya que para las tramas con un tamaño por encima del umbral siempre está activado el mecanismo pero para las tramas con un tamaño inferior presenta un comportamiento semejante al de AARF-CD.

Un hecho importante es que cada vez que se produce una nueva conexión a un nuevo punto de acceso o reasociación el algoritmo de selección de tasa reinicia su comportamiento, comenzando de nuevo desde la tasa física más baja. Este hecho debe tenerse en cuenta en los algoritmos de roaming ya que, aunque el nuevo AP sea capaz de ofrecernos un mayor caudal de tráfico, será necesario un tiempo para poder recuperar la tasa de transmisión que la estación está empleando en el enlace que tiene con el AP actual.

## **4.2. Algoritmos de asociación y roaming**

Como ya se ha comentado anteriormente el objetivo de este trabajo es estudiar las posibles alternativas que se pueden emplear a la hora de asociar y reasociar las STAs con uno u otro punto de acceso. Analizando estos resultados se elaborará, como conclusión final, unos algoritmos de decisión mixta que realicen una ponderación de varias métricas. El objetivo final de todo esto debe ser conseguir balancear de la forma más óptima posible la carga de las STAs consiguiendo el máximo throughput global posible del despliegue y con ello una mayor QoS.

Todos los algoritmos cuentan con unos márgenes o umbrales que determinarán en qué condiciones se aprobará realizar un traspaso del actual AP a otro y en su implementación hay que distinguir entre asociaciones (la STA no está conectada a ningún AP) y reasociaciones (la STA va a realizar un handover) ya que los criterios no son los mismos.

Así pues, las métricas que se van a estudiar como variable para balancear las cargas entre los diferentes son seis. La mayoría se puede emplear en despliegues reales si los sistemas soportan el estándar 802.11k (ver anexo F para más información). Estos criterios de decisión se pueden dividir en dos grupos según el objetivo que pretenden conseguir: optimizar las condiciones de canal o balancear la carga.

### 4.2.1. Algoritmos de optimización del canal radio

Dentro de este grupo se pueden diferenciar tres métricas: la distancia entre la STA y el AP, la potencia de recepción y la relación de señal a ruido más interferencia.

#### 1. Distancia

- Descripción: Esta métrica corresponde a la distancia en línea recta existente entre el nodo móvil y el AP. Este valor se encuentra más o menos considerado en el estándar 802.11k aunque sin la misma precisión milimétrica usada en el simulador.
- Objetivo: Este algoritmo hace que las STAs se asocien al AP más cercano por lo que no pretende balancear la carga en ningún momento, sino que se está escogiendo la opción que presentará en teoría las menores pérdidas de propagación.
- Criterio de reasociación: Solo se aprobará el traspaso al AP<sub>objetivo</sub> si la diferencia entre la distancia del AP objetivo y el actual es superior a un determinado valor ( $\text{Margen}_{\text{Distancia}}$ ). De esta forma, se evita que una misma STA sufra el efecto rebote en las reasociaciones.

$$\text{Distancia}_{AP\text{ objetivo}} \leq \text{Distancia}_{AP\text{ actual}} - \text{Margen}_{\text{Distancia}} \quad (4.1)$$

#### 2. Potencia recibida (RSSI) en los Beacon

- Descripción: Esta métrica representa el valor medio de potencia con la que se han recibido los Beacon. A pesar de no ser un criterio estandarizado es posiblemente la variable que más se emplea en la decisión de que AP escoger en los despliegues reales. Esta métrica se encuentra en el estándar 802.11k en las mismas condiciones que las empleadas en la simulación aunque no es necesario que las STAs soporten esta versión para poder emplearla en las asociaciones.
- Objetivo: Este algoritmo tampoco pretende balancear la carga sino que su objetivo es maximizar la potencia tanto en el downlink como en el uplink para poder optar a transmitir con el mejor MCS posible y por lo tanto maximizar el throughput tanto de la STA como del escenario global.
- Criterio de reasociación: En este caso la medida se encuentra fuertemente influenciada por la presencia de shadowing por lo que es necesario introducir un margen de histéresis ( $\text{Margen}_{\text{RSSI}}$ ) que impida que se produzca el efecto ping-pong, es decir, reasociaciones demasiado consecutivas debido al efecto del fading en los handover de las STAs.

$$\text{RSSI}_{AP\text{ objetivo}}[\text{dB}] \geq \text{RSSI}_{AP\text{ actual}}[\text{dB}] + \text{Margen}_{\text{RSSI}}[\text{dB}] \quad (4.2)$$

#### 3. SINR calculada en los Beacon

- Descripción: Esta métrica representa la relación señal a ruido más interferencia media que la STA ha calculado a lo largo de la recepción del Beacon. Es importante indicar que los niveles de SINR para el downlink y uplink son diferentes, basándose esta métrica exclusivamente en el downlink. En la realidad los resultados de esta variable difieren mucho de los obtenidos por el criterio anterior al existir la interferencia entre canales. Al igual que la potencia, también se encuentra implementada directamente tanto en la capa de gestión de los RRM definida en el primer estándar 802.11 como en los posibles campos opcionales de las tramas expandidas en el 802.11k.

- Objetivo: Presenta exactamente el mismo objetivo que la anterior, asociarse al AP que permita emplear el mejor MCS posible.
- Criterio de reasociación: Al igual que en el caso del RSSI, es necesario introducir un margen de histéresis ( $\text{Margen}_{\text{SINR}}$ ) que impida que se produzca el efecto ping-pong. En este caso, además de los efectos del shadowing, el margen también deberá mitigar los posibles efectos perniciosos que la interferencia cocanal pudiera provocar.

$$\text{SINR}_{AP\text{objetivo}}[dB] \geq \text{SINR}_{AP\text{actual}}[dB] + \text{Margen}_{\text{SINR}}[dB] \quad (4.3)$$

En todos estas métricas el criterio de asociación implica exclusivamente la selección del AP que se encuentra a una menor distancia o del que se recibe una mayor RSSI o SINR respectivamente.

## 4.2.2. Algoritmos de balanceo del tráfico

### 1. Número de estaciones asociadas al AP

- Descripción: Esta métrica consiste en que los nodos se asocian al punto de acceso que tiene menos número de estaciones asociadas más un pequeño margen que se comentará posteriormente. El empleo de esta variable se contempla en las mismas condiciones que las usadas en este trabajo en el estándar 802.11k.
- Objetivo: Con este algoritmo lo que se consigue es balancear el número de usuarios asociadas a cada AP sin importar el tráfico que las STAs están transmitiendo o incluso si se encuentran activas en ese momento.
- Criterio de reasociación: Puesto que el espíritu de esta métrica es balancear el número de STAs asociadas a cada AP, el margen óptimo para conseguirlo es de 2. Un valor inferior desestabilizaría el escenario ya que el cambio de un único nodo podría conllevar el cambio de otros muchos. Adicionalmente, es necesario otro criterio más para aprobar el traspaso y es que la potencia de recepción de la trama Beacon sea superior a un determinado umbral ( $\text{RSSI}_{\text{min}}$ ) para evitar que la STA realice un traspaso a un AP con el que las condiciones de canal son realmente desfavorables.

$$N^{\circ} \text{Stas}_{AP\text{objetivo}} \leq N^{\circ} \text{Stas}_{AP\text{actual}} + \text{Margen}_{N\text{Sta}} \quad (4.4)$$

$$\text{RSSI}_{AP\text{objetivo}}[dB] \geq \text{RSSI}_{\text{min}}[dB]$$

### 2. Tasa cursada por el AP

- Descripción: Esta variable representa el tráfico que ha sido cursado por el punto de acceso durante los últimos segundos. En el estándar 802.11k no se encuentra definida explícitamente esta variable pero si hay otras parecidas como la capacidad de admisión disponible estimada para cada una de las categorías de tráfico y prioridades de usuario definidas en 802.11e.
- Objetivo: Al contrario que todas las anteriores métricas, esta si tiene como objetivo principal balancear el tráfico entre los distintos puntos de acceso. Así, el algoritmo

buscará que la tasa cursada por los puntos de acceso sea lo más equitativa posible, balanceando con ello el número de bytes que son transmitidos en cada celda.

- Criterio de reasociación: En este caso se ha planteado la necesidad de imponer la satisfacción de dos criterios de decisión para que el algoritmo tuviera un comportamiento lo más adecuado posible. Un primer criterio impone que el cociente entre el tráfico cursado por el AP objetivo y el AP actual se sitúe por encima de un determinado umbral ( $Umbral_{Ratio\ tasa}$ ). El segundo criterio impone un valor mínimo de tasa ( $Umbral_{Inferior\ tasa}$ ) que deberá estar cursándose en nuestro actual punto de acceso para considerar el roaming. Adicionalmente se ha vuelto a introducir un umbral mínimo en la potencia ( $RSSI_{min}$ ) con la que se debe recibir el Beacon para evitar traspasos a APs con unas condiciones de canal muy desfavorables.

$$Tasa\ cursada_{AP\ objetivo} \leq Tasa\ cursada_{AP\ actual} \cdot Umbral_{Ratio\ tasa} \quad (4.5)$$

$$Tasa\ cursada_{AP\ actual} \geq Umbral_{inferior\ tasa}$$

$$RSSI_{AP\ objetivo} [dB] \geq RSSI_{min} [dB]$$

### 3. Carga del canal radio detectada por el AP

- Descripción: Esta métrica representa el porcentaje del tiempo que el mecanismo CCA del AP ha detectado que el canal radio está ocupado. Los motivos para que esté ocupado pueden ser varios: el primero es que el AP esté transmitiendo. El segundo es que haya detectado que una STA o incluso otro AP están transmitiendo una trama en el mismo canal. El último es que el nivel de ruido, sin importar su procedencia, sea demasiado elevado. Las fuentes de ruido pueden ser: interferencias ajenas al sistema Wi-Fi, por estaciones que transmiten en el mismo canal pero están demasiado lejos o por la interferencia entre canales. En el estándar 802.11k podemos encontrar definida esta medida exactamente igual que la que se ha utilizado en el trabajo.
- Objetivo: Esta variable es, de todas las planteadas, la que mejor permite balancear la carga, que no maximizar el throughput, ya que la medida sí que es una representación fidedigna del empleo que se está haciendo de los recursos radio y en consecuencia de la capacidad de transmisión que queda disponible para aceptar nuevas peticiones de tráfico.
- Criterio de reasociación: Al igual que en el criterio anterior se ha optado por imponer la satisfacción de dos criterios que permitan que las reasociaciones no desestabilicen el escenario. El primero de ellos ( $Umbral_{Ratio\ carga}$ ) es una diferencia en términos absolutos del porcentaje de ocupación del canal del AP objetivo con el AP actual y el segundo ( $Umbral_{Inferior\ carga}$ ) es el valor mínimo de ocupación que debe superarse en el punto de acceso en el que la STA está asociado para comenzar el handover. El nivel mínimo de potencia recibida en el Beacon ( $RSSI_{min}$ ) vuelve a imponerse para evitar traspasos a APs con unas condiciones de canal desfavorables.

$$Carga\ canal_{AP\ objetivo} \leq Carga\ canal_{AP\ actual} - Umbral_{ratio\ carga} \quad (4.6)$$

$$Carga\ canal_{AP\ actual} \geq Umbral_{inferior\ carga}$$

$$RSSI_{AP\ objetivo} [dB] \geq RSSI_{min} [dB]$$

En los algoritmos de balanceo de tráfico (número de usuarios, tasa cursada y carga del canal), el criterio de asociación implica exclusivamente la selección del AP que cuenta con el menor número de usuarios conectados, la menor tasa cursada y la menor carga del canal respectivamente.

En las reasociaciones se ha introducido la potencia en forma de umbral para evitar que las STAs realicen traspasos en condiciones de canal realmente desfavorables, provocando con ello una reasociación a los pocos segundos en muchos casos al perder la conexión con el nuevo AP. Además, hay que considerar que una reasociación en uno de estos tres algoritmos tiene efectos inmediatos en el resto de STAs ya que las métricas de decisión están influenciadas por la situación y el tráfico que genera cada una de las estaciones. En cuanto al valor de -90 dBm que se ha escogido para este umbral se ha realizado de forma completamente experimental ya que tras unas cuantas simulaciones se pudo observar que era el valor más adecuado al minimizar el número de handovers y maximizar el throughput.

En la tabla 4.1 se muestran los posibles umbrales se van a considerar y sobre los que se van a realizar algunas simulaciones de prueba para buscar sus valores óptimos.

Métrica	Variable	Posibles valores	
Distancia	Margen <sub>Distancia</sub>	5, 10, 15	m
RSSI	Margen <sub>RSSI</sub>	1, 3, 5, 7	dB
SINR	Margen <sub>SINR</sub>	1, 3, 5, 7	dB
Nº STAs	Margen <sub>NSTA</sub>	2	usuarios
Tasa cursada	Umbral <sub>Ratio tasa</sub>	20, 30, 40, 50	%
	Umbral <sub>Inferior tasa</sub>	1; 1,25; 1,5; 1,75	Mbps
Carga canal	Margen <sub>Ratio carga</sub>	10, 20, 30	%
	Umbral <sub>Inferior carga</sub>	30, 40, 50, 60	%

Tabla 4.1: Métricas y sus posibles márgenes/umbrales de reasociación

Para más información sobre estas métricas y otras que no se han implementado por diferentes motivos se puede consultar el anexo D, donde además también se encuentra una explicación mucho más detallada de la implementación que ha sido necesaria para la obtención de estas variables en el simulador y los algoritmos que las emplean para decidir las asociaciones y traspasos, incluyendo ejemplos comentados de código.

### 4.2.3. Algoritmos mixtos

Por último, se propone el empleo de algoritmos de decisión que no estén basados en una única métrica, sino que combinen varias (en este caso se ha elegido emplear el RSSI y la carga del canal). Juntando las bondades de ambas se intentará que la carga se balancee de la mejor forma a la vez que se mantengan las mejores condiciones del canal posibles, logrando con ello una mayor

optimización de los recursos. La ponderación de estos dos parámetros será uno de los resultados del trabajo ya que se estudiarán diferentes alternativas.

#### 4.2.3.1 Algoritmo mixto 1

Para desarrollar este primer algoritmo ha sido necesario conocer las probabilidades de error que poseen cada una de las modulaciones que se encuentran modeladas en el simulador ns-3 [25] [26] y que se emplean en el estándar 802.11g. En la figura 4.1 se puede ver el modelo para las modulaciones OFDM heredadas de 802.11a y las modulaciones DSSS/CCK procedentes de la norma 802.11b.

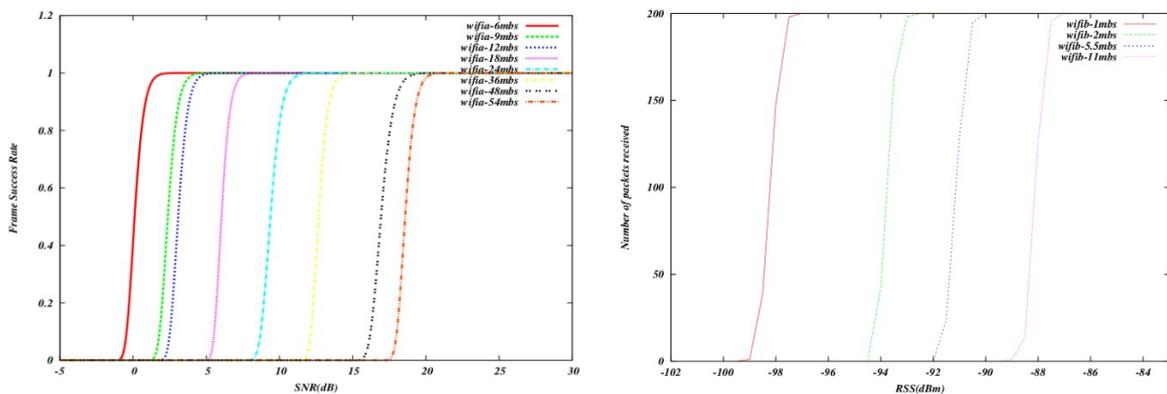


Figura 4.1: Modelo de errores de las modulaciones OFDM (802.11a/g) y DSSS(802.11b/g) en Yans

Observando las figuras anteriores se pueden extraer los valores de SINR o RSSI que son la frontera entre una modulación y otra, pudiendo establecer unas ventanas de SINR o RSSI óptimas de uso para cada una de ellas. A cada una de estas ventanas se les ha añadido un margen ( $\text{Margen}_{\text{seguridad}}$ ) para desplazar las ventanas hacia valores más seguros y se les ha asignado un peso ( $\text{Peso}_{\text{RSSI}}$ ) que en este caso corresponde con el de la tasa física de transmisión de cada modulación. En cuanto a los posibles valores de carga del canal también se establecen 8 posibles ventanas de valores a las que se les asigna un peso ( $\text{Peso}_{\text{carga}}$ ). (ver tabla 4.2)

Finalmente, la métrica elegida es el producto de los pesos asociados a ambos parámetros. Adicionalmente, y como en el resto de algoritmos, se aplica un factor ( $\text{Margen}_{\text{mixto1}}$ ) en la comparación de las métricas para penalizar los traspasos ya que, como se ha comentado con anterioridad, el cambio de AP reinicia el algoritmo de selección de tasa.

Las ecuaciones (4.7) y (4.8) son las expresiones algebraicas que se evalúan para realizar el algoritmo. En el anexo D se encuentra su implementación en el simulador ns-3.

$$\text{Pesototal} = \text{Peso}_{\text{RSSI}} \cdot \text{Peso}_{\text{carga}} \quad (4.7)$$

$$\text{Pesototal}_{\text{APobjetivo}} \geq \text{Pesototal}_{\text{APactual}} \cdot \text{Margen}_{\text{mixto1}} \quad (4.8)$$

Peso RSSI	Ventana de decisión		Peso carga	Ventana de decisión	
	Mínimo	Máximo		Mínimo	Máximo
1	$-96 + \text{Margen}_{\text{seguridad}}$	$-93 + \text{Margen}_{\text{seguridad}}$	1	90%	100%
2	$-93 + \text{Margen}_{\text{seguridad}}$	$-91 + \text{Margen}_{\text{seguridad}}$	2	80%	90%
6	$-91 + \text{Margen}_{\text{seguridad}}$	$-88 + \text{Margen}_{\text{seguridad}}$	3	70%	80%
9	$-88 + \text{Margen}_{\text{seguridad}}$	$-87,5 + \text{Margen}_{\text{seguridad}}$	4	60%	70%
12	$-87,5 + \text{Margen}_{\text{seguridad}}$	$-84,5 + \text{Margen}_{\text{seguridad}}$	5	50%	60%
18	$-84,5 + \text{Margen}_{\text{seguridad}}$	$-81 + \text{Margen}_{\text{seguridad}}$	6	40%	50%
24	$-81 + \text{Margen}_{\text{seguridad}}$	$-78 + \text{Margen}_{\text{seguridad}}$	7	30%	40%
36	$-78 + \text{Margen}_{\text{seguridad}}$	$-73,5 + \text{Margen}_{\text{seguridad}}$	8	0%	30%
48	$-73,5 + \text{Margen}_{\text{seguridad}}$	$-72 + \text{Margen}_{\text{seguridad}}$			
54	$-72 + \text{Margen}_{\text{seguridad}}$	-			

Tabla 4.2: Pesos del algoritmo mixto 1

En cuanto al valor de los márgenes se realizaron múltiples simulaciones con diferentes configuraciones y número de usuarios, obteniéndose que los valores más adecuados eran Margen seguridad = 2 y Margen mixto1 = 2,75.

El resultado de aplicar la tabla 4.2 con el margen de seguridad ya incluido es la figura 4.2, donde se puede observar la relación que existe entre la potencia RSSI y la ocupación del canal en este primer algoritmo de decisión mixta.

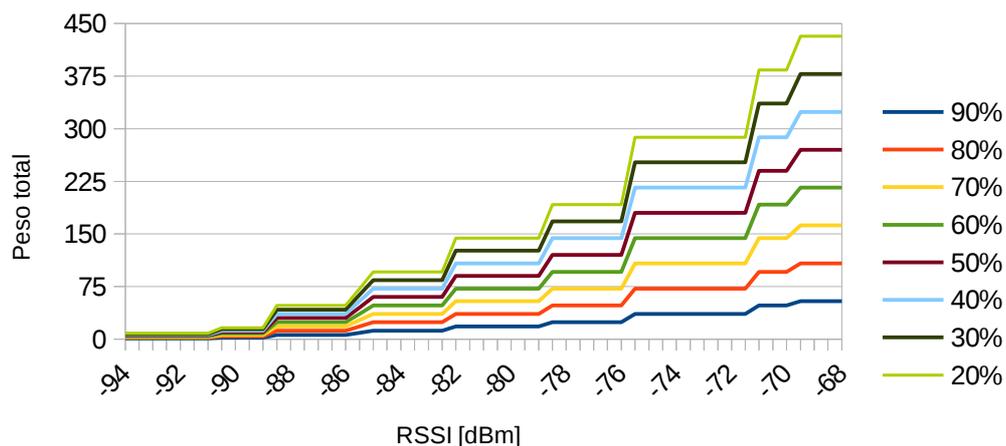


Figura 4.2: Relación ocupación del canal y RSSI en el algoritmo mixto 2

### 4.2.3.2 Algoritmo mixto 2

El algoritmo mixto 2 consiste en una ponderación casi directa de los valores obtenidos de potencia recibida y de carga de canal (ecuación 4.9). En este caso, se calcula el excedente de potencia y de ocupación del canal y ambos son multiplicados por unos pesos ( $\alpha_{RSSI}$  y  $\alpha_{carga}$ , respectivamente). Finalmente se aplica otro margen ( $Margen_{mixto2}$ ) para penalizar el roaming (ecuación 4.10).

$$Peso\ total = \alpha_{RSSI} \cdot (RSSI_{dBm} - Sensibilidad_{dBm}) + \alpha_{carga} \cdot (100 - Carga\ canal) \quad (4.9)$$

$$Peso\ total_{AP\ objetivo} \geqslant Peso\ total_{AP\ actual} + Margen_{mixto2} \quad (4.10)$$

De nuevo, en el anexo D se encuentra el código que se ha implementado para incorporar este algoritmo en el simulador.



## 5. Resultados

Para la obtención de los resultados se han seguido tres pasos. El primero ha sido establecer los umbrales óptimos para las diferentes métricas por separado. El segundo ha consistido en comparar los mejores resultados de los diferentes algoritmos. El tercero y último consiste en evaluar la mejora que supone el empleo de un algoritmo de asociación mixto frente a los anteriores.

### 5.1. Condiciones de evaluación

Respecto a la generación del tráfico se ha optado por que las estaciones generadoras sean las mismas STAs y todo se dirija a un nodo conectado a la red LAN que interconecta a todos los AP. El motivo de elegir que todo el tráfico sea unidireccional en el sentido del uplink es que al, ir dirigido todo el tráfico a un mismo nodo, es mucho más fácil de monitorizar todas las pérdidas y el número de paquetes que se han conseguido transmitir de forma satisfactoria.

Los parámetros de generación de tráfico se recopilan en la tabla 5.1. Cada STA genera sesiones siguiendo un proceso de Poisson con duración exponencial. En cada sesión, para la aplicación generadora de datos se ha optado por una fuente con un patrón On-Off. La tasa que se indica corresponde con la tasa de generación de tráfico en el periodo de ON.

Duración periodo ON	400, 500	ms	dist. exponencial
Duración periodo OFF	400, 500	ms	dist. exponencial
Tamaño paquete	172, 512, 1024, 2048	bytes	constante
Tasa generada	64, 1000, 2000	kbps	constante
Duración conexiones	10	s	dist. exponencial
Tiempo entre conexiones	10	s	dist. exponencial

Tabla 5.1: Parámetros de generación de tráfico

La distribución y la dirección del movimiento de las STAs se han determinado de forma completamente aleatoria, estando únicamente fijada la velocidad a 4km/h.

Para que la distribución de las STAs sea lo más homogénea posible en el escenario, este se ha dividido en 9 cuadrados de mismo tamaño en cuyo centro se encontrará un punto de acceso. Cada simulación tendrá como parámetro de entrada el número de estaciones medio que tendrá cada una de estas zonas. El número final de estaciones en el despliegue será un número obtenido de una

variable aleatoria con una distribución uniforme con el valor medio introducido y una desviación de  $\pm 2$  usuarios.

En cuanto a la dirección del movimiento se ha optado por emplear un desplazamiento rectilíneo de dirección variable. Cada 9 segundos la estación cambia de dirección en un rango de  $-90$  a  $90$  grados (ver figura 5.1). Los 9 segundos están relacionados con la velocidad de movimiento y la velocidad con la que cambia la variable aleatoria del shadowing, escogiendo que la estación cambiara de dirección cada 10 metros. Para aleatorizar la dirección de movimiento en el instante inicial se ha usado una variable aleatoria de 0 a 359 para calcular el ángulo que tendrá el desplazamiento rectilíneo uniforme (mientras no alcance los límites del escenario) y de  $-90$  a  $90$  en los sucesivos cambios de dirección. Para evitar que las estaciones se salieran del escenario se ha implementado un sistema de rebotes en el que las STAs se reflejan manteniendo siempre su velocidad y variando únicamente la dirección.

El motivo de que se haya implementado un movimiento aleatorio de estas características es que es más realista que un movimiento donde la dirección varíe de forma aleatoria, lo que podría provocar que una estación apenas variara su posición a lo largo del tiempo.

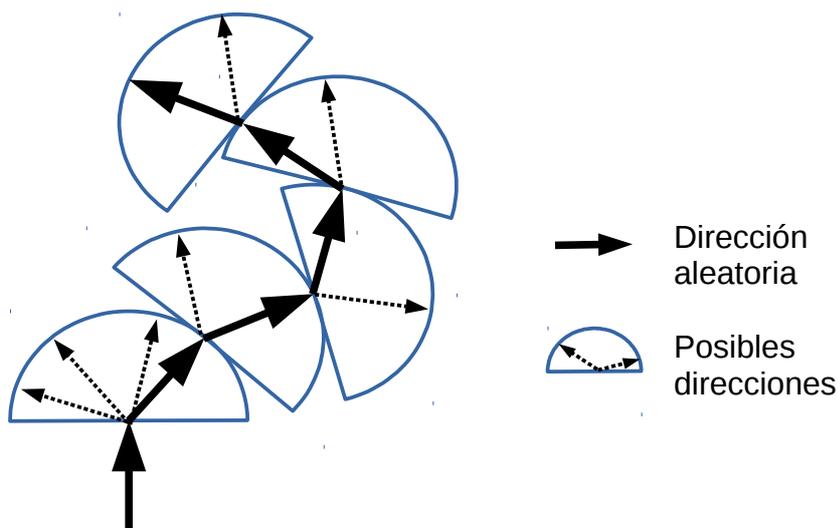


Figura 5.1: Movimiento rectilíneo variable

En cuanto a la configuración de las simulaciones, duración de estas y el número de simulaciones necesarias con los mismos parámetros aleatorios, los valores se encuentran resumidos en la tabla 5.2.

Duración simulación total	120	segundos
Duración transitorio	20	segundos
Duración estacionario	100	segundos
Nº simulaciones/escenario	30	simulaciones

Tabla 5.2: Parámetros generales de la simulación

La comparación de los diferentes criterios se va a realizar atendiendo al throughput global medio alcanzado, el retardo medio de transmisión y el porcentaje de pérdidas en distintas condiciones de carga.

Para entender los resultados hay que tener en cuenta la siguiente consideración. Cuando se habla de número de usuarios medio se está haciendo referencia a los que se han localizado inicialmente en cada una de las 9 zonas en las que se ha dividido el escenario tal y como se ha explicado en el capítulo anterior.

## **5.2. Umbrales experimentales**

En primer lugar es necesario conocer los umbrales óptimos que maximizan el rendimiento de los diferentes algoritmos planteados en el apartado anterior. Para realizar las pruebas se han empleado una tasa intermedia (1 Mbps) que permita tener unas situaciones sin congestión y otras con elevadas pérdidas debido a la falta de recursos. En cuanto al tamaño al tamaño de paquete se ha escogido un paquete de 1024 bytes que es un valor intermedio situado entre el máximo MPDU posible (2.304 bytes) y los valores típicos de las aplicaciones en tiempo real como la VoIP o los paquetes de videojuegos (por ejemplo, 160 bytes).

En las siguientes gráficas se pueden observar los resultados que se obtuvieron con diferentes números de usuarios en el escenario, un tamaño de paquete de 1024 bytes y una tasa de 1000kbps.

### **5.2.1. Umbral RSSI/SINR**

A pesar de que los resultados obtenidos para el criterio de RSSI y de SINR no sean exactamente los mismos, las diferencias no suponen ninguna diferencia en el valor del margen óptimo. Así, las siguientes gráficas se corresponden con los valores obtenidos (para el throughput y el retardo) en las simulaciones donde se ha empleado la RSSI como criterio de traspaso.

Aunque en la bibliografía hay comparaciones interesantes [8], estas se realizaron en escenarios diferentes al de este trabajo, por lo que en esta primera evaluación se incluye una comparación de los tres algoritmos de control de tasa introducidos en el capítulo anterior.

Observando los resultados (ver figuras 5.2 y 5.3), se puede ver que el mejor algoritmo de adaptación de tasa para un escenario como el planteado es CARA, por lo que a partir de ahora todos los resultados corresponderán con los obtenidos de las simulaciones que han usado este algoritmo.

El comportamiento de AARF-CD es muy semejante al de CARA y solamente ve reducido el throughput global en un escaso 2-3% de media y un ligero aumento del retardo de transmisión, siendo estas diferencias despreciables en los casos con menor número de estaciones y, por lo tanto, menor congestión.

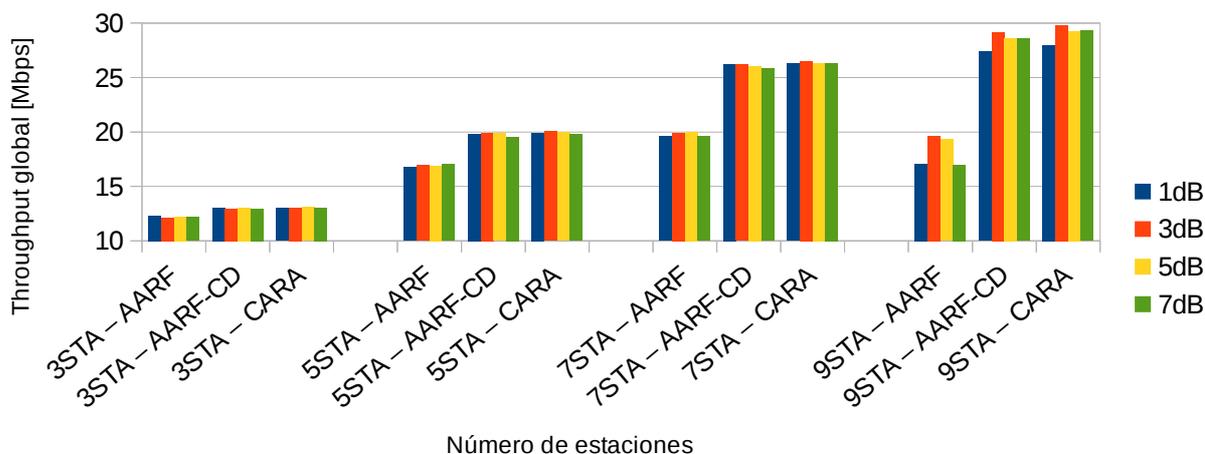


Figura 5.2: Throughput global del escenario – Márgenes RSSI/SINR

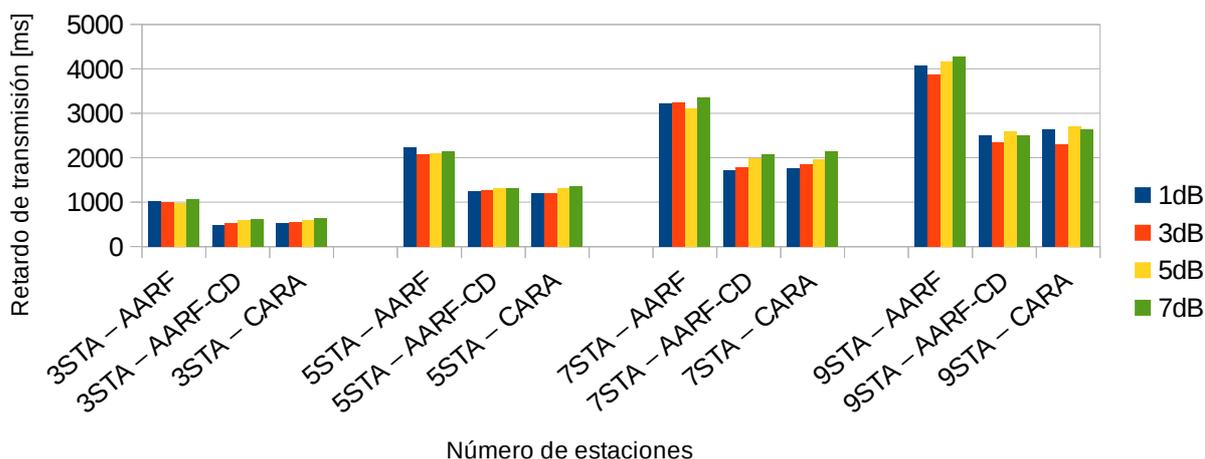


Figura 5.3: Retardo de transmisión medio – Márgenes RSSI/SINR

En cuanto a las enormes diferencias que existen con el algoritmo AARF con el mecanismo RTS/CTS desactivado, estas se deben a que el algoritmo es incapaz de conocer si las pérdidas se están produciendo por colisiones o por un MCS inadecuado, por lo que todas las estaciones acaban transmitiendo a la menor tasa posible. Esto se traduce en una reducción drástica del rendimiento cuanto mayor es la demanda de recursos, dándose la circunstancia de que el throughput global con 7 STAs es mayor que con 9 STAs a pesar de que el tráfico generado es menor.

Centrándonos en el algoritmo CARA, el valor óptimo de  $\text{Margen}_{\text{RSSI}}$  para decidir realizar un handover se encuentra en los 3dB. Aunque en las situaciones con poca carga en el sistema (número medio de STAs bajo) se comporta de una forma muy semejante al resto, cuando hay saturación se obtienen unos mejores resultados debido a que un umbral pequeño como este permite que haya cierta flexibilidad en las estaciones para realizar un traspaso a la vez que no retrasa demasiado el momento del handover.

### 5.2.2. Umbral distancia

Las figura 5.4 y 5.6 muestran los resultados de throughput, retardo y número de traspasos obtenidos para distintos márgenes de distancia. El margen de distancia ( $\text{Margen}_{\text{Distancia}}$ ) debe ser lo suficientemente grande para compensar los posibles movimientos de las estaciones pero no tan excesivo como para evitar que los traspasos se retrasen lo suficiente para que las condiciones de canal se degraden demasiado.

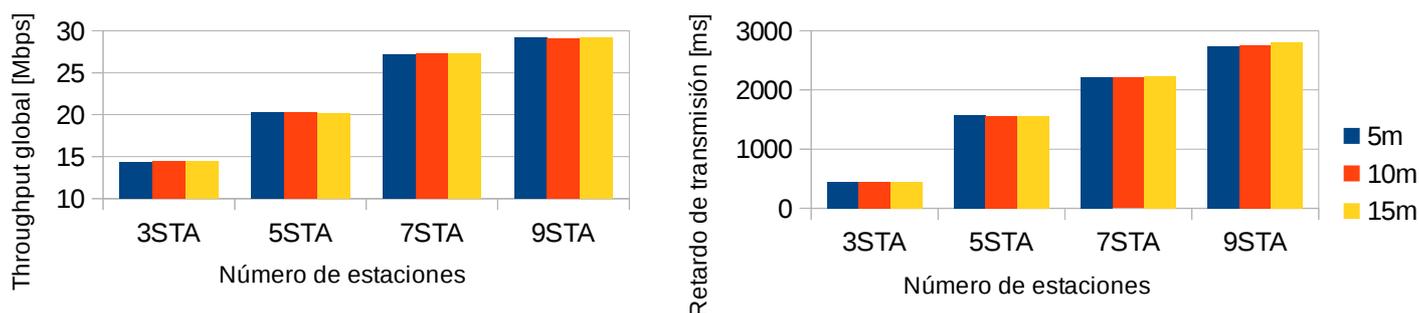


Figura 5.4: Throughput global y retardo de transmisión medio – Márgenes distancia

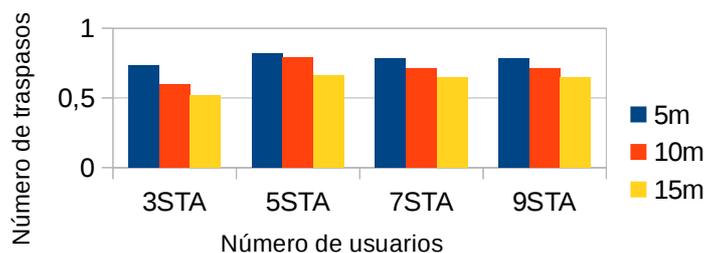


Figura 5.5: Número de traspasos – Márgenes distancia

De los valores anteriores se puede observar que el comportamiento es muy parecido para los distintos márgenes. Observando la figura 5.5, se puede deducir el motivo de esta similitud ya que cuando se produce un traspaso, el algoritmo de tasa comienza desde la tasa más baja, por lo que la ventaja de estar siempre más cerca del AP al usar un umbral más bajo se pierde al perder también la tasa de transmisión al reiniciarse el algoritmo de selección de tasa. Finalmente se ha escogido el valor intermedio, 10 metros.

### 5.2.3. Umbral tasa cursada

Al contrario que en los dos anteriores criterios, para caracterizar por completo el algoritmo que emplea la tasa cursada como métrica de decisión es necesario realizar un mayor número de simulaciones ya que cuenta con dos criterios de decisión. Los resultados de throughput y retardo (ver figuras 5.6 y 5.7) se obtienen para distintos ratios ( $\text{Umbral}_{\text{Ratio tasa}} = 20, 30, 40$  y  $50\%$ ) y umbrales ( $\text{Umbral}_{\text{Inferior tasa}} = 1; 1,25, 1,5$  y  $1,75\text{Mbps}$ ).

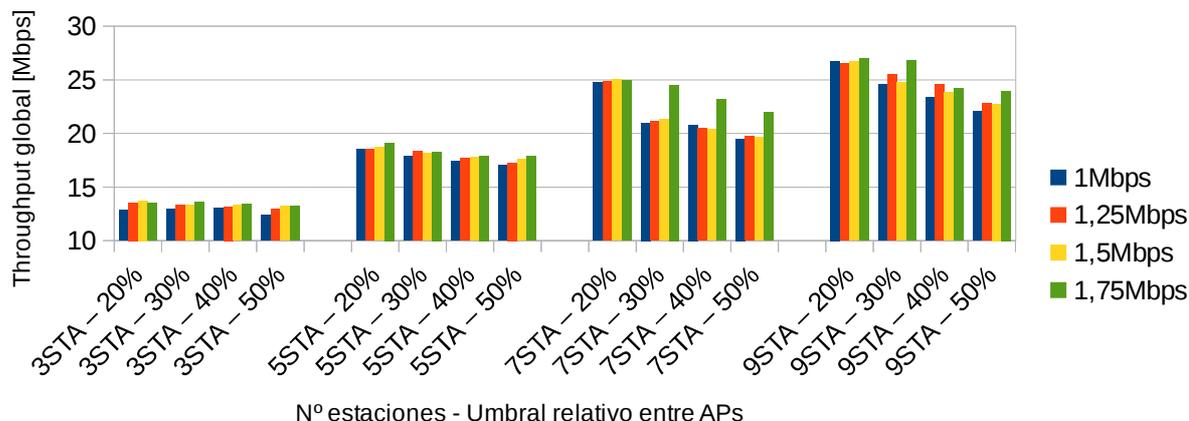


Figura 5.6: Throughput global del escenario – Umbrales tasa cursada

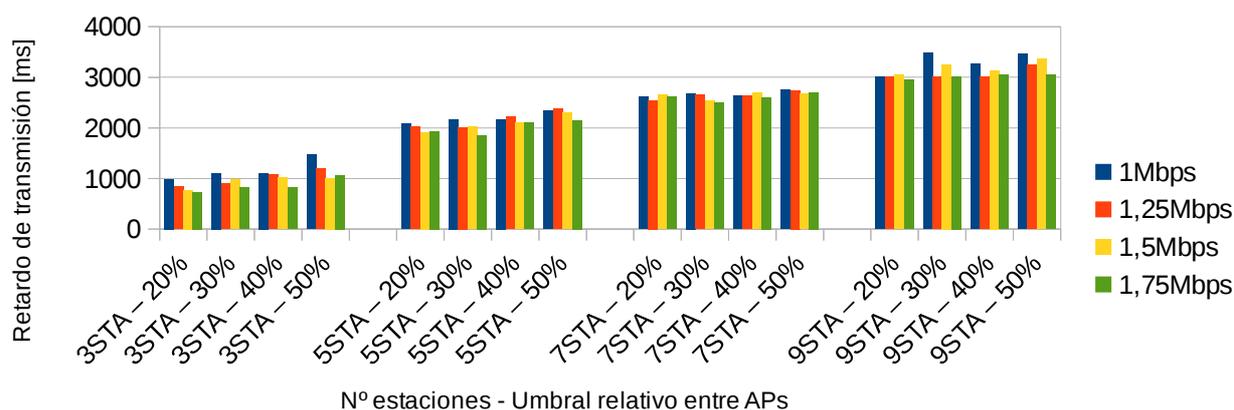


Figura 5.7: Retardo de transmisión medio – Umbrales tasa cursada

Lo primero que destaca en las figuras 5.6 y 5.7 es que el impacto que tiene la elección de unos umbrales respecto a otros es mucho más grande que en los dos casos anteriores, mientras que no hay unos valores que se comporten mejor en todas las situaciones (con pocos o muchos usuarios).

Sin embargo, observando detenidamente se puede ver que un umbral inferior ( $\text{Umbral}_{\text{Inferior tasa}}$ ) de 1,75 Mbps y un umbral relativo ( $\text{Umbral}_{\text{Ratio tasa}}$ ) del 20% es la combinación que presenta un mejor comportamiento en la mayoría de las situaciones, obteniendo un throughput global superior al resto en todos los escenarios y degradándose únicamente un poco la latencia en algunas circunstancias. La peculiaridad de que el umbral relativo sea tan bajo y el umbral mínimo tan alto es que ambos valores permiten que el número de traspasos sea lo más bajo posible, manteniendo con ello la tasa física que se está empleando.

### 5.2.4. Umbral carga del canal

Al igual que en el caso anterior, con la métrica de carga del canal también es necesario realizar un mayor número de simulaciones al tener dos criterios de decisión. Los resultados de throughput y retardo de transmisión (ver figuras 5.8 y 5.9) se obtienen para distintos ratios ( $\text{Margen}_{\text{Ratio carga}} = 10, 20 \text{ y } 30\%$ ) y umbrales ( $\text{Umbral}_{\text{Inferior carga}} = 30; 40, 50 \text{ y } 60\%$ ).

El valor del umbral inferior ( $\text{Umbral}_{\text{Inferior carga}}$ ) debe ser lo suficientemente alto para evitar que se produzcan traspasos cuando la carga del canal en nuestro actual AP es lo suficientemente baja como para que un cambio de AP no suponga una ventaja pero no tan alto que retrase el roaming. En cuanto a  $\text{Margen}_{\text{Ratio carga}}$ , su valor deberá ser lo suficientemente alto como para que el traspaso suponga una mejora y lo suficientemente bajo como para que no se retrase demasiado el handover.

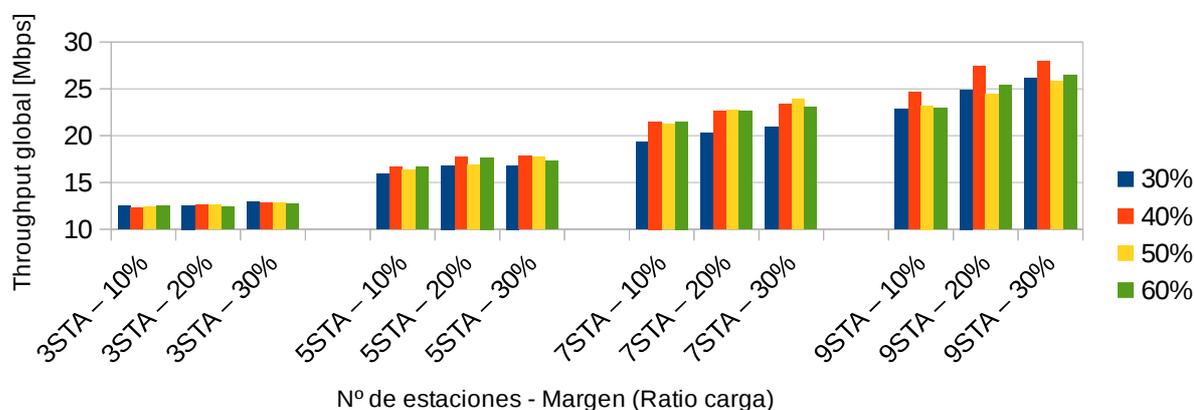


Figura 5.8: Throughput global del escenario – Umbrales carga del canal

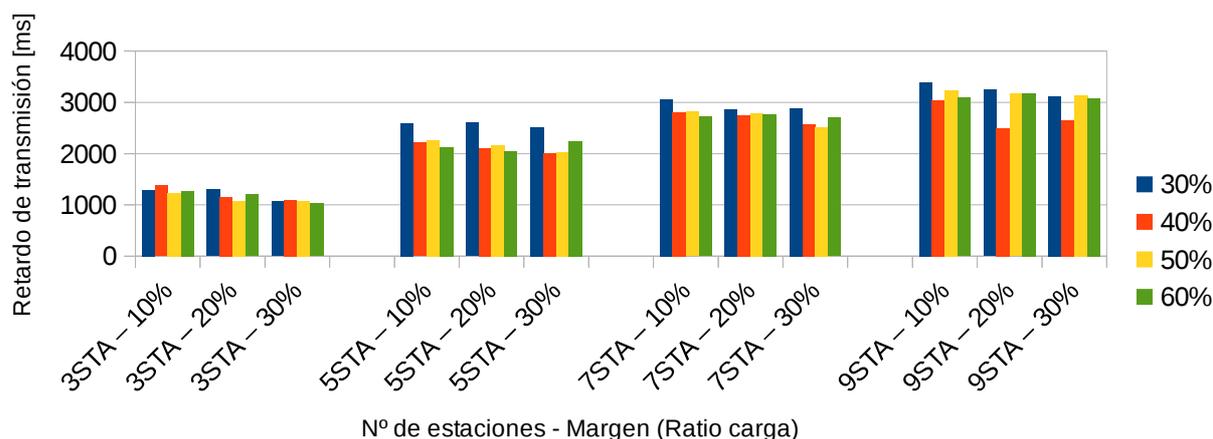


Figura 5.9: Retardo de transmisión medio – Umbrales carga del canal

Observando las figuras 5.8 y 5.9 se puede ver que los márgenes y umbrales más extremos presentan un peor comportamiento. En el caso en los que hay pocas estaciones, y en consecuencias poco tráfico, el throughput es muy parecido en todos los casos ya que al haber tan poca carga todos

los márgenes y umbrales presentan un comportamiento muy similar. Sin embargo, cuando el tráfico aumenta, se puede ver que la combinación de valores ( $\text{Margen}_{\text{Ratio carga}}=20\%$  y  $\text{Umbral}_{\text{Inferior carga}}=40\%$ ) presenta un mejor comportamiento que el resto tanto en throughput como en el retardo de transmisión, por lo que estos son los valores escogidos para este criterio de selección.

### 5.3. Evaluación algoritmos básicos

Todos los valores que se han considerado óptimos en el apartado anterior y que por lo tanto se van a emplear en las simulaciones finales son los que se encuentran en la tabla 5.3. Además, también se ha decidido que el mejor algoritmo de control de tasa es CARA.

Métrica	Variable	Valor	
Distancia	$\text{Margen}_{\text{Distancia}}$	10	m
RSSI	$\text{Margen}_{\text{RSSI}}$	3	dB
SINR	$\text{Margen}_{\text{SINR}}$	3	dB
Nº STAs	$\text{Margen}_{\text{NSTA}}$	2	usuarios
Tasa cursada	$\text{Umbral}_{\text{Ratio tasa}}$	20	%
	$\text{Umbral}_{\text{Inferior tasa}}$	1,75	Mbps
Carga canal	$\text{Margen}_{\text{Ratio carga}}$	30	%
	$\text{Umbral}_{\text{Inferior carga}}$	40	%

Tabla 5.3: Métricas y sus umbrales de reasociación definitivos

En las dos siguientes figuras se encuentran en una misma gráfica todos los anteriores algoritmos con sus umbrales óptimos (ver tabla 5.3) para poder comparar de un simple vistazo las bondades y defectos que presenta cada uno.

Observando únicamente el throughput global y la latencia (ver figura 5.10) ya se puede comprobar que los algoritmos cuyo objetivo es proporcionar a la STA las mejores condiciones de canal posible se comportan mejor en todas las condiciones, tanto cuando hay pocas estaciones como en las que hay las suficientes para congestionar el canal. La SINR se presenta como la mejor alternativa ya que, teniendo un comportamiento muy parecido a la RSSI, mejora sus prestaciones cuando hay un mayor número de STAs debido a que en su cálculo se tiene en cuenta la interferencia generada por el resto de estaciones y que al haber mucho tráfico, es un elemento a tener en cuenta puesto que degrada la calidad del canal de las estaciones. En cuanto a los criterios de balanceo de tráfico, los criterios de número de estaciones y tasa cursada presentan el mismo comportamiento. El algoritmo que emplea la carga del canal, como ya era de esperar, mejora las prestaciones cuando el escenario se encuentra saturado ya que reparte la carga entre los diferentes puntos de acceso, empleando lo mejor posible los recursos disponibles.

Respecto al retardo de transmisión el comportamiento es muy semejante al throughput global. Únicamente comentar que la métrica de distancia presenta unos retardos superiores a los algoritmos de RSSI y SINR. Esto se puede deber, como ya se ha comentado anteriormente, a que a pesar de que la estación se encuentra enganchado al AP que está más cercano, las condiciones del canal radio no son las mejores posibles. Además, al igual que en el throughput, el criterio de carga de canal mejora mucho sus prestaciones cuando hay mayor tráfico.

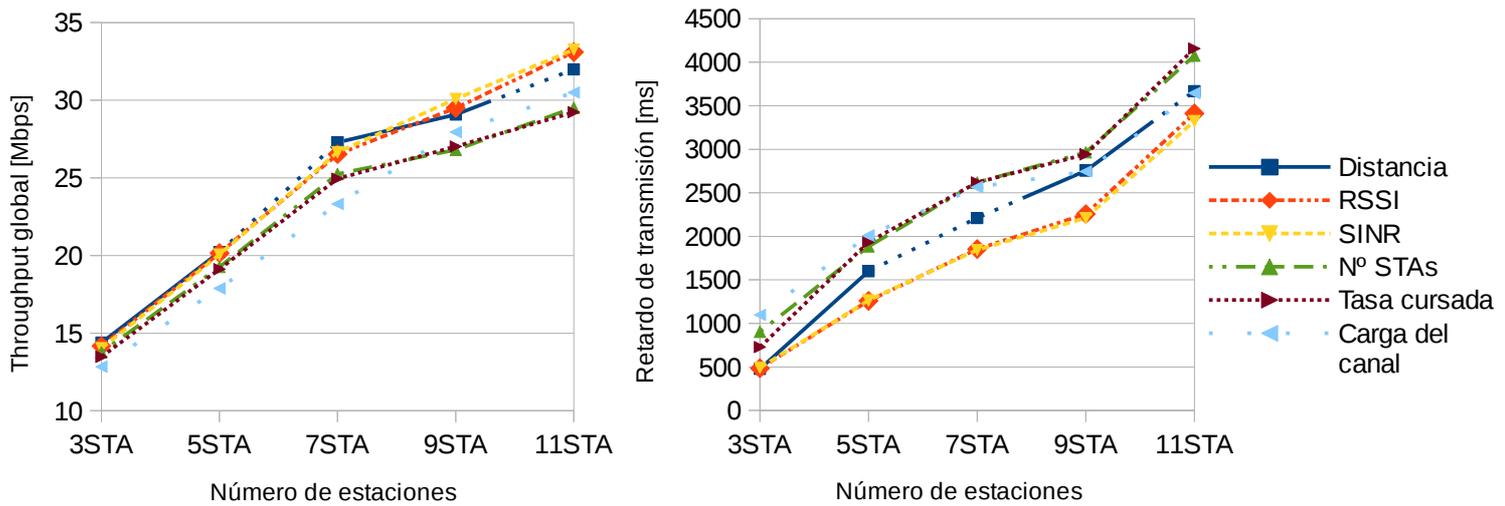


Figura 5.10: Throughput global y retardo de transmisión medio – Comparativa criterios – 1Mbps/1024B

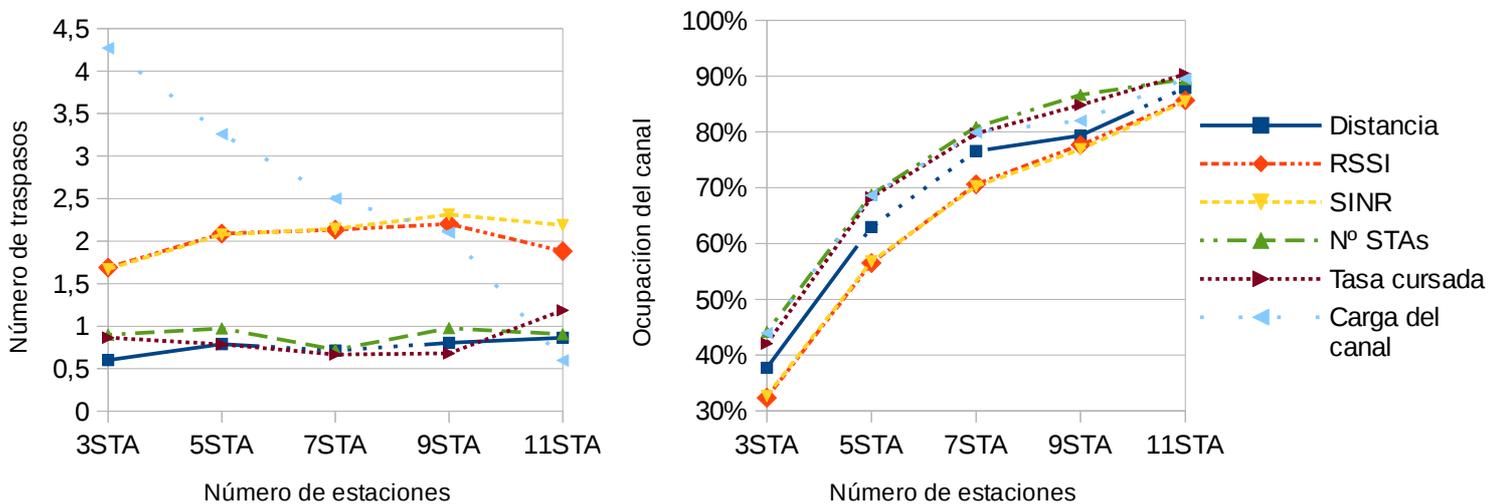


Figura 5.11: Número de traspasos y ocupación del canal – Comparativa criterios – 1Mbps/1024B

En la figura 5.11 se pueden ver otras estadísticas que caracterizan de una forma más precisa los criterios: el número de traspasos medio y la ocupación del canal media estimada en los puntos de acceso. En el primer caso se puede ver como únicamente el criterio de carga de canal se ve afectado por el número de estaciones, decayendo el número de reasociaciones cuanto más tráfico

hay. El comportamiento es estable para el resto de criterios independientemente del número de estaciones presentes en el despliegue debido a que se está analizando el sistema en su fase permanente y omitiendo el transitorio, donde las métricas de tasa cursada o de número de estaciones presentan un comportamiento más inestable.

En cuanto a la ocupación del canal se produce un efecto que llama la atención, los criterios que más ocupan el canal no son los que consiguen una mayor tasa de transmisión. Esto se debe a que los algoritmos que balancean la carga realizan un mejor uso y distribución de los recursos radio disponibles pero la velocidad de transmisión media es más baja. Así, el canal está ocupado pero con unas tasas de transmisión menores que en el caso de los criterios de distancia, RSSI o SINR.

A partir de todas estas observaciones parece claro que un algoritmo que combine varias métricas de una forma adecuada deberá presentar el buen comportamiento que tienen las métricas que optimizan el canal cuando hay poco tráfico y un mejor balance del tráfico entre las distintas celdas del despliegue.

## 5.4. Evaluación algoritmos mixtos

Antes de comparar los algoritmos mixtos, fue necesario establecer los valores óptimos de sus márgenes y umbrales. En las dos siguientes figuras se encuentra la caracterización del algoritmo mixto 1, donde se han probado diferentes valores de sus parámetros  $\text{Margen}_{\text{seguridad}}$  (1, 2, 3 dB) y  $\text{Margen}_{\text{mixto1}}$  (2; 2,5 y 3).

Al igual que en la caracterización de las métricas anteriores, en las siguientes gráficas se pueden observar los resultados que se obtuvieron con diferentes números de usuarios en el escenario, un tamaño de paquete medio (1024 bytes) y una tasa de 1000kbps.

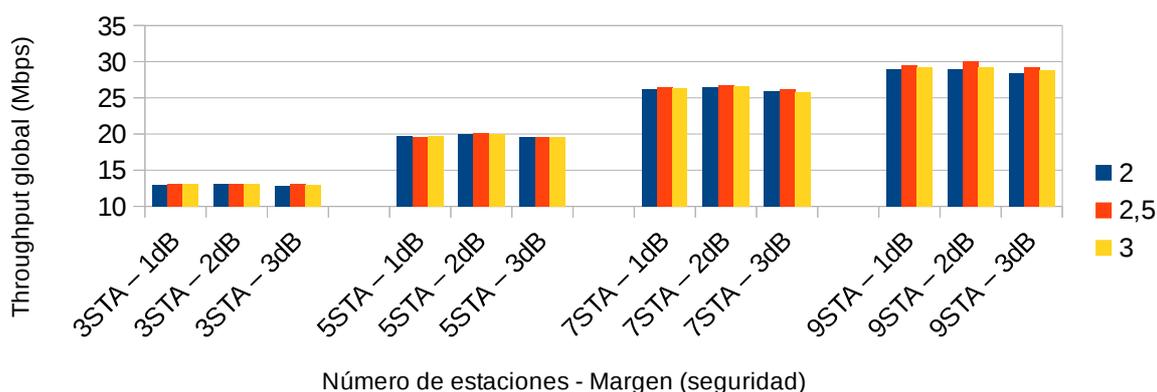


Figura 5.12: Throughput global del escenario – Márgenes algoritmo mixto1

En el caso de encontrarnos con poca carga en el sistema, todas configuraciones se comportan de una forma muy semejante (ver figuras 5.12 y 5.13). Cuando hay un mayor número de estaciones en el despliegue es cuando se empiezan a producir algunas diferencias tanto en el

throughput global como en el retardo de transmisión. Así, la mejor combinación es:  $\text{Margen}_{\text{seguridad}}=2\text{dB}$  y  $\text{Margen}_{\text{mixto1}}=2,5$ .

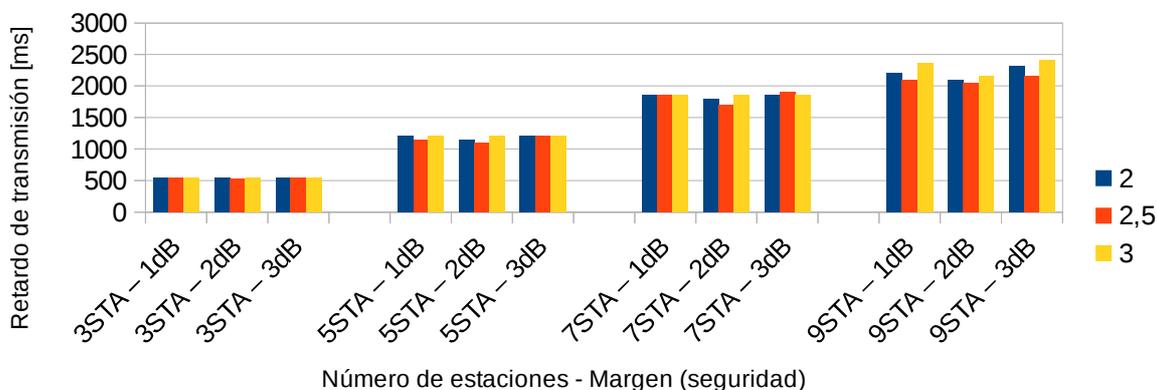


Figura 5.13: Retardo de transmisión medio – Márgenes algoritmo mixto1

Para el algoritmo mixto 2, se puede fijar un peso de la ponderación y encontrar el valor más adecuado del otro. En este caso, se fija  $\alpha_{\text{RSSI}}=10$ . En las figuras 5.14 y 5.15 se encuentran los resultados de realizar varias simulaciones con diferentes valores de  $\alpha_{\text{carga}}$  (1; 1,2; 1,4) y  $\text{Margen}_{\text{mixto2}}$  (1; 1,5; 2).

En un primer momento se realizaron simulaciones con los pesos con valores similares pero los resultados no eran satisfactorios porque otorgaban demasiada importancia a la ocupación del canal. En el caso de que  $\alpha_{\text{carga}}$  sea superior a  $\alpha_{\text{RSSI}}$  se produce el mismo problema. Así, los mejores resultados se obtuvieron cuando el peso del RSSI es muy superior al de carga de canal.

A la vista de los resultados obtenidos, las figuras 5.14 y 5.15 muestran únicamente las simulaciones donde ya se ha acotado la muestra entorno a los mejores valores. Observando ambas gráficas, resulta claro que la mejor combinación de valores es  $\alpha_{\text{carga}}=1,2$  y  $\text{Margen}_{\text{mixto2}}=1,5$ . Estos valores presenta un mejor comportamiento tanto en condiciones de baja carga como en escenarios saturados.

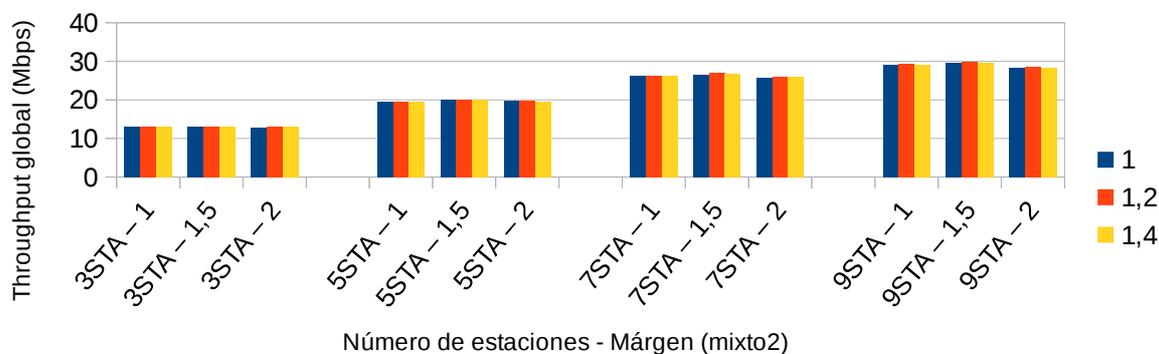


Figura 5.14: Throughput global del escenario – Umbrales algoritmo mixto2

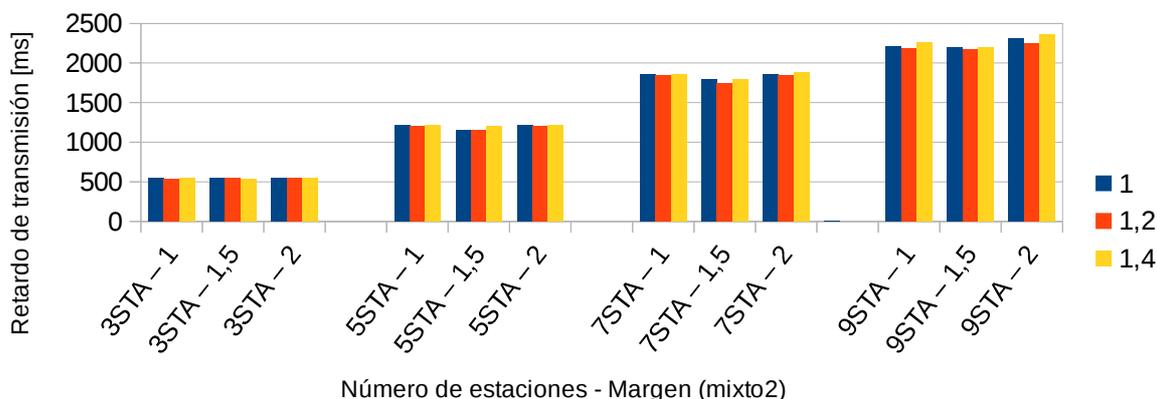


Figura 5.15: Retardo de transmisión medio – Umbrales algoritmo mixto2

En las siguientes gráficas se pueden observar los resultados de throughput y retardo que se obtuvieron con diferentes números de usuarios en el escenario, un tamaño de paquete grande de 2048bytes y uno más pequeño de 512bytes y unas tasas de 1 o 2 Mbps en el periodo de ON.

En estas dos primeras figuras ya se puede observar que ambos algoritmos propuestos mejoran el comportamiento de todos los criterios de selección anteriores. En el caso de que el tráfico no llegue a congestionar el canal, los dos nuevos criterios de selección se comportan de una forma similar a los criterios RSSI o SINR que optimizan el MCS que se puede emplear. Cuando el tráfico aumenta y se produce congestión, los dos nuevos algoritmos mejoran las prestaciones al conseguir reducir tanto el retardo de transmisión como balancear mejor la carga entre las distintas celdas del despliegue.

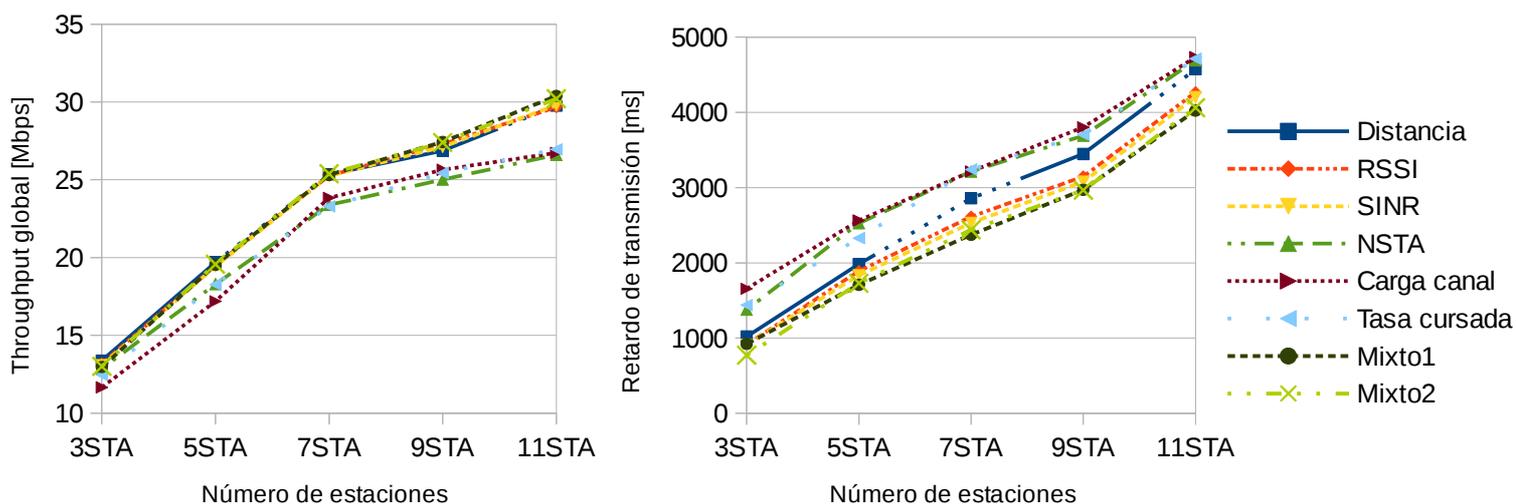


Figura 5.16: Throughput global y retardo de transmisión medio – Comparativa criterios – 1Mbps/512B

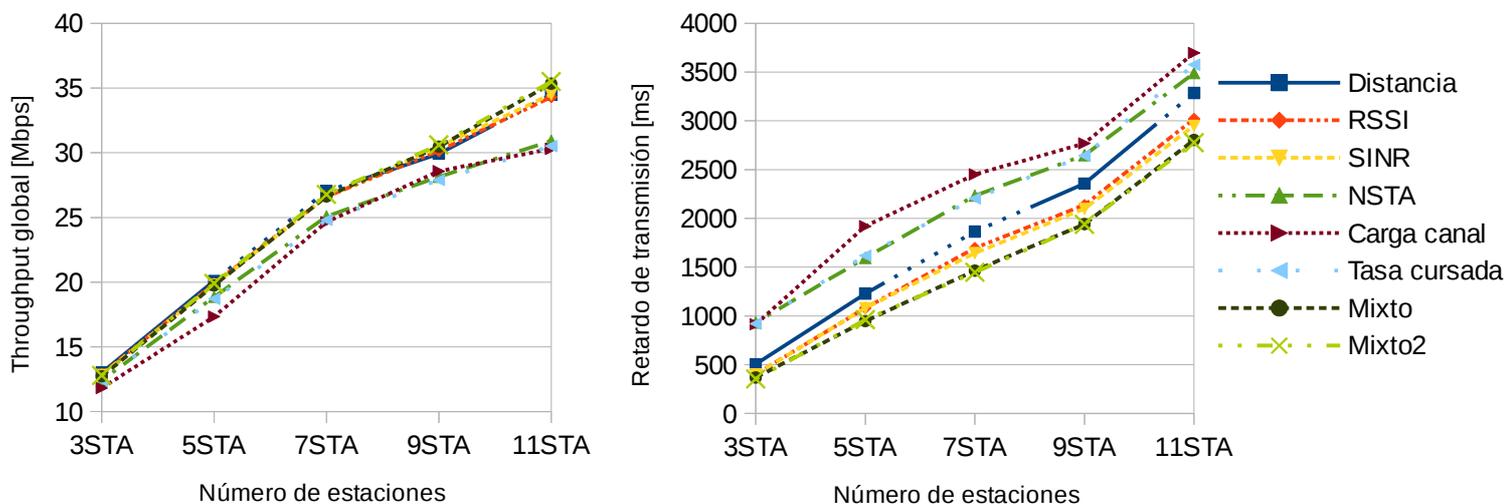


Figura 5.17: Throughput global y retardo de transmisión medio – Comparativa criterios - 1Mbps/2048B

Además, como ya se comentó anteriormente, el impacto del tamaño del paquete no es despreciable ya que, con una misma tasa de generación de tráfico y por el único hecho de estar empleando un tamaño de paquete mayor, se está consiguiendo un 17% más de throughput global. En el caso del retardo de transmisión la mejora todavía es más evidente ya que el retardo medio es un 77% superior cuando se emplea un tamaño de paquete pequeño. El motivo de estas pérdidas es tanto el overhead introducido por las cabeceras de nivel MAC y nivel físico que se introducen en las tramas como por el overhead introducido por el protocolo de acceso al medio CSMA/CA.

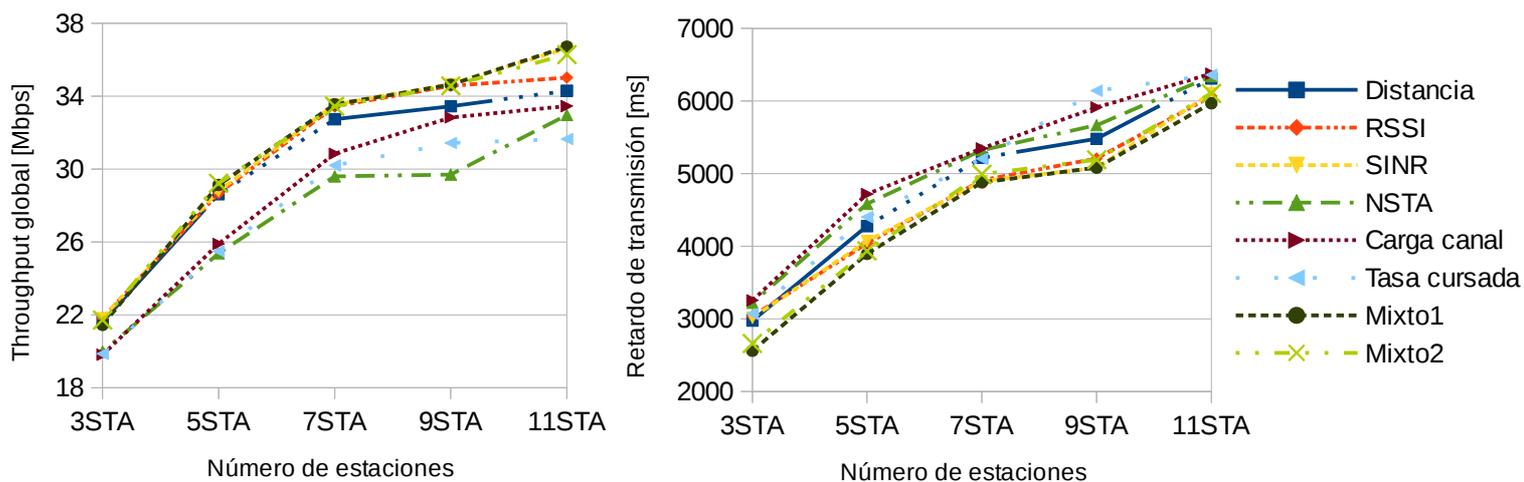


Figura 5.18: Throughput global y retardo de transmisión medio – Comparativa criterios – 2Mbps/512B

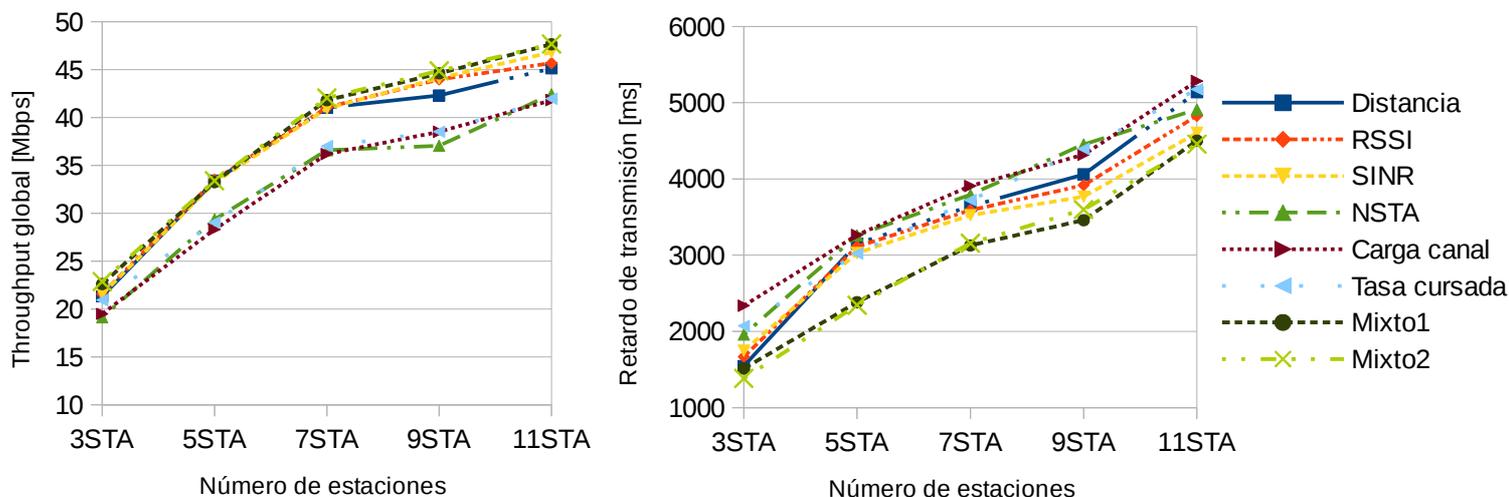


Figura 5.19: Throughput global y retardo de transmisión medio – Comparativa criterios – 2Mbps/2048B

En las anteriores gráficas (ver figuras 5.18 y 5.19) se encuentran los resultados de emplear una mayor tasa de transmisión, y en consecuencia, escenarios con un mayor nivel de congestión. Como se puede comprobar en las figura 5.18, los nuevos criterios de decisión mixta presentan una mayor mejora en su comportamiento respecto a los criterios simples cuanto mayor es la carga en el sistema.

En este caso, las mejoras por emplear paquetes de un mayor tamaño todavía son mayores ya que, en las situaciones con mayor tráfico, las mejoras son superiores al 32% en el throughput global y de un 100% en el retardo de transmisión para los algoritmos de decisión mixta propuestos en este trabajo.

A partir de todos estos resultados, en tráficos donde el retardo de transmisión no es importante como una conexión FTP, resulta muy interesante la agregación de tramas que introduce el estándar 802.11n. Este mecanismo permite reducir el overhead y disminuir el tiempo de transmisión entre tramas de la misma transmisión.

Finalmente se encuentra la simulación con el paquete más pequeño, de 172 bytes, 160bytes de nivel de aplicación + 12bytes de RTP (*Real Time Protocol*), y 64 kbps de generación de tráfico. Este tamaño y velocidad se han escogido para simular el comportamiento que presentaría la red ante la presencia de un tráfico similar al de VoIP con un vocoder estandarizado como el G.711 [27]. En este caso, al contrario que los anteriores, no se interesa medir el throughput global sino que hay que revisar la tasa de pérdidas de paquetes para cumplir que el límite que establece el estándar ya que por encima del 1% de paquetes perdidos la calidad de la llamada decae drásticamente.

Para realizar estas simulaciones se ha modificado el comportamiento de los buffer, modificando el tiempo tras el que son descartados los paquetes. En este caso se ha considerado que un retardo de 50ms en la red de acceso es una cota razonable.

La figura 5.20 representa el retardo de transmisión y la tasa de pérdidas de paquetes. Como en los casos anteriores, el comportamiento de los algoritmos que pretenden únicamente balancear la carga es muy malo y netamente inferior a los que optimizan las condiciones de canal. Únicamente los dos algoritmos mixtos y el de SINR son capaces de mantener por debajo del 1% las pérdidas cuando el número de usuarios medio por celda se sitúa en 9.

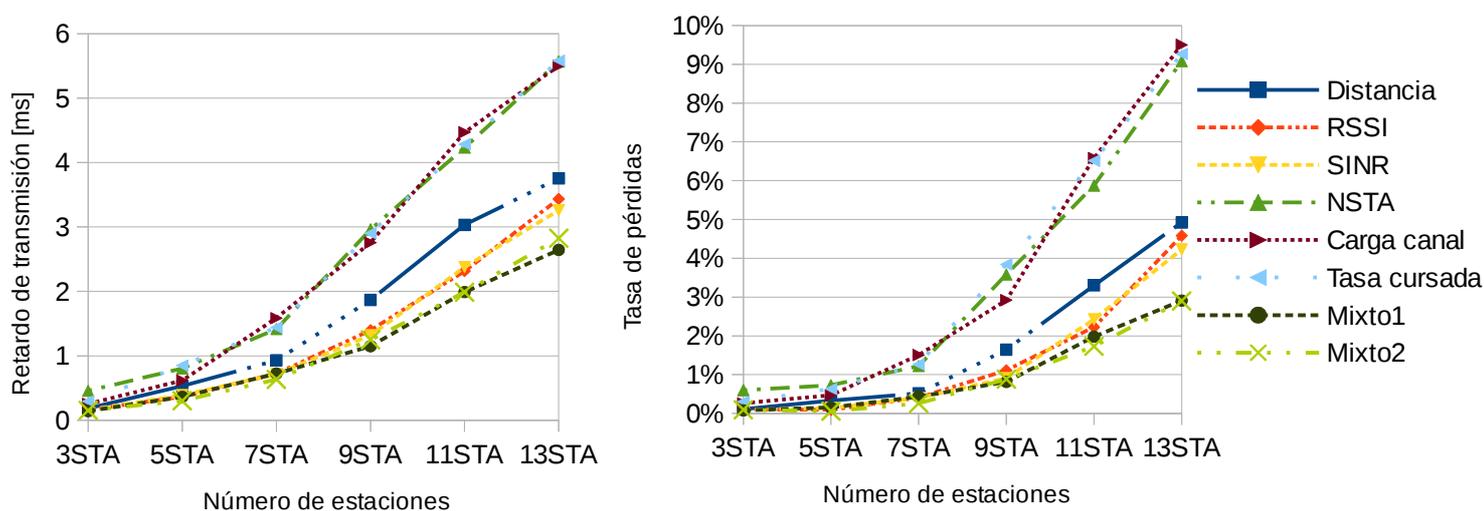


Figura 5.20: Retardo de transmisión y tasa de pérdidas de paquetes – Comparativa criterios – 64kbps/172B

Un dato interesante es la dependencia que tiene el retardo de transmisión con la tasa de pérdidas. A pesar de que el retardo medio es muy inferior en todas las situaciones a los 50ms establecidos como cota, la tasa de pérdidas se dispara en cuanto aumenta el número de usuarios. Esto se debe a que los paquetes que son encolados o son transmitidos en poco tiempo (con bajo retardo) o directamente son descartados.



## 6. Conclusiones y líneas futuras

### 6.1. Conclusiones

El objetivo de este proyecto ha sido estudiar las posibles alternativas existentes en la gestión de los recursos radio a través de la evaluación de diferentes criterios en la selección de los puntos de acceso en despliegues de área local.

Gran parte del tiempo invertido que inicialmente no se tenía previsto se ha empleado en la compresión de la estructura del software simulador ns-3 y en la implementación de diversas funcionalidades como el soporte multicanal que se han considerado necesarias para conseguir unos resultados más realistas. También se han modelado las condiciones de canal con un modelo de propagación lo más realista posible incluyendo pérdidas por shadowing.

A la par que se adaptaba el comportamiento del simulador a las exigencias que requería este trabajo se configuró el escenario con parámetros reales extraídos tanto de estándares, como valores regulados o procedentes de dispositivos reales.

A partir de algunas propuestas de la bibliografía y de los mecanismos ya existentes en los normas IEEE 802.11 se ha procedido a evaluar diferentes métricas a partir de las cuales se pueden establecer algoritmos de selección o reelección de puntos de acceso para optimizar el canal radio y balancear la carga de las diferentes estaciones.

Otro elemento que se ha comprobado ser extremadamente importante para mejorar la provisión de QoS en los despliegues ha sido el algoritmo de adaptación tasa que escogamos ya que el empleo de un algoritmo respecto a otro puede suponer una ganancia de más del 30% en la velocidad de transmisión media.

Finalmente, a partir de todos los resultados que se han podido extraer de la simulación, se han elaborado dos algoritmos de selección y reelección de puntos de acceso con lo que se ha conseguido mejorar las prestaciones del sistema al conseguir balancear la carga entre los distintos APs de la WLAN.

La valoración personal del trabajo es muy satisfactoria ya que a pesar de los numerosos inconvenientes que han ido surgiendo conforme se comprendía la estructura y las limitaciones del simulador, se han podido solventar todos con esfuerzo y dedicación. Solo lamentar que el elevado tiempo invertido en configurar y comprender el simulador no se haya podido invertir en estudiar un mayor número de configuraciones.

## 6.2. Líneas futuras

Las posibles continuaciones naturales de este trabajo pueden ir encaminadas hacia tres sentidos: continuar con la implementación en ns-3, seguir con el mismo modelo pero ampliar las pruebas y verificar en un escenario real la veracidad de los resultados obtenidos

En primer lugar, tras dedicar tantas horas a comprender la estructura y el funcionamiento del simulador ns-3 uno se da cuenta de las enormes posibilidades que ofrece y del enorme potencial que tiene gracias a la enorme comunidad que tiene detrás contribuyendo activamente al proyecto tanto depurando los modelos existentes como desarrollando nuevas funcionalidades. Así, las dos primeras líneas futuras de este trabajo serían las dos siguientes:

- Serializar el código para contribuir con la comunidad

La necesidad de tener un modelo que soporte varios canales, aunque sean sin solape espectral, y cuyas STAs tengan soporte de roaming se antoja necesario para poder realizar simulaciones que se asemejen lo más posible a los despliegues actuales y en este trabajo se ha dado solución a este problema con la implementación de un modelo que ha arrojado unos resultados satisfactorios.

Sin embargo, el modelo que se ha construido para soportar múltiples canales Wi-Fi en una misma estación de forma simultánea y transparente para el resto de nodos cuenta con algunas particularidades que, para acortar el tiempo de desarrollo del código, no permiten que las clases tengan un comportamiento aislado, sino que necesitan algunos datos del script de configuración del escenario para la inicialización de variables o funciones. Con invertir un poco más de tiempo se podría serializar el código y contribuir a la comunidad con dos nuevas funcionalidades: el soporte multicanal e itinerancia, con diferentes criterios, en las estaciones móviles.

- Implementar el protocolo 802.11k en ns3

Relacionado con el punto anterior y para no alargar todavía más la implementación, algunas métricas se han guardado de una forma que no corresponde con la situación que ocurriría en los despliegues reales ya que estas métricas no se están intercambiando realmente a través de los mensajes Beacon y Probe. Así, una continuación natural del trabajo sería modificar el contenido de las tramas de gestión según sugiere el estándar 802.11k para que incluyeran en su cuerpo las medidas solicitadas por las estaciones.

Como ya se ha comentado anteriormente, la elevada duración de las simulaciones que crece exponencialmente con la introducción de nuevos nodos y nuevo tráfico junto con la limitada duración de este trabajo no han permitido probar todos los escenarios que inicialmente se habían planeado. Algunas líneas futuras serían las siguientes:

- Continuar probando más configuraciones

Evaluar de una forma más extensa el impacto de los tamaños, tasas de generación de paquetes, factor de actividad, emplear conexiones que se encapsulen bajo TCP, aumentar el número de APs y STAs, modificar la disposición de los puntos de acceso, incluir la banda de

5GHz, modificar el estándar 802.11 empleado... Evaluar los esquemas en condiciones de distribución más heterogénea de tráfico en el despliegue es otra opción. Son muchos los parámetros que se podrían estudiar si el trabajo tuviera una mayor duración.

- Introducción de QoS y usuarios con diferentes aplicaciones  
Aprovechando que el simulador ya tiene implementado la priorización de usuarios y tráfico a través de las colas definidas en el estándar de QoS 802.11e se podrían introducir estaciones con diferentes necesidades de tráfico y estudiar algoritmos que optimizaran la provisión de la calidad de servicio.
- Incorporar adaptación de potencia  
Incluir la adaptación de potencia en los puntos de acceso para permitir que las áreas de cobertura se adaptaran dinámicamente para balancear la carga y las estaciones según las demandas de tráfico.

Finalmente, el último paso podría ser implementar los algoritmos de asociación en un despliegue real y realizar pruebas de campo para poder verificar si los resultados obtenidos se mantienen en condiciones reales. La familia de routers WRT54G de la marca Linksys podrían ser un buen entorno donde poder implementar estas soluciones debido a la gran facilidad que presentan para introducirles un firmware con las capacidades requeridas y a que una de sus versiones, la WRT54GL, tiene disponible todo el código fuente al ser desarrollado bajo Linux.



## ANEXOS



## Anexo A. Acrónimos

- AAC: Available Admission Capacity
- ACK: Acknowledgement
- AIFS: Arbitration Interframe Space
- AP: Access Point
- ARP: Address Resolution Protocol
- BSS: Basic Service Set
- BSSID: Basic Service Set Identifier
- CCA: Clear Channel Assessment
- CCK: Complementary Code Keying
- CSMA/CA: Carrier Sense Multiple Access / Collision Avoidance
- CSMA/CD: Carrier Sense Multiple Access / Collision Detection
- CTS: Clear to Send
- CW: Contention Windows
- DCF: Distributed Coordination Function
- DIFS: DCF Interframe Space
- DLS: Direct Link Setup
- DS: Distribution System
- DSSS: Direct Sequence Spread Spectrum
- EDCA: Enhanced distributed channel access
- EIFS: Extended Interframe space
- ERP: Extended Rate Physical
- ESS: Extended Service Set
- FIFO: First In, First Out
- FTTx: Fiber To The x
- HC: Hybrid Coordinator
- HCCA: HCF Controlled Channel Access

- HFC: Hybrid Fibre Coaxial
- IAPP: Inter-Access Point Protocol
- IEEE: Institute of Electrical and Electronics Engineers
- IP: Internet Protocol
- ISM: Industrial, Scientific and Medical
- LLC: Logical Link Control
- MAC: Media Access Control
- MCS: Modulation and Coding Scheme
- MIB: Management Information Base
- MSB: Most Significant Bits
- MTU: Maximum Transfer Unit
- MLME: MAC Layer Management Entity
- NAV: Network Allocation Vector
- NS: Network Simulator
- OFDM: Orthogonal Frequency Division Multiplexing
- PCF: Point Coordination Function
- PDU: Protocol Data Unit
- PIFS: PCF Interframe Space
- PLCP: Physical Layer Convergence
- PLME: Phy Layer Management Entity
- PMD: Physical Medium Dependent
- RCPI: Received Channel Power Indicator
- RIFS: Reduced Interframe Space
- RRM: Radio Resource Management
- RSSI: Received Signal Strength Indicator
- RTS: Request to Send
- SAP: Service Access Point
- SIFS: Short Interframe Space
- SINR: Signal to Interference plus Noise Ratio

- SME: Station Management Entity
- SSID: Service Set Identifier
- STA: Station
- QoS: Quality of Service
- UDP: User Data Protocol
- VoIP: Voice over IP
- WLAN: Wireless Local Area Network
- WAN: Wide Area Network



# Anexo B. Introducción a Network Simulator 3

## B.1 Introducción

Network Simulator 3 [28], abreviado comúnmente como ns-3, es un simulador de eventos discretos de código libre que es ampliamente empleado para propósitos educativos y de investigación. Es un software de código libre y se distribuye bajo licencias GNU GPLv2 que permite disponer de todo el código del programa para su uso y modificación tanto para uso personal como comercial.

En la siguiente tabla se puede ver de forma resumida las principales diferencias entre ambas versiones y las ventajas que presenta ns-3 frente a ns-2 [29].

	ns-2	ns-3
Última versión	2.35 (4-11-2011)	3.23 (14-5-2015)
Lenguajes	C++, oTcl (scripts)	C++, Python (opcional)
Mantenimiento	El soporte de la comunidad se está reduciendo.	La comunidad que contribuye al proyecto sigue aumentando en la actualidad.
Rendimiento	La compilación de oTcl toma menos tiempo que la de C++.	En simulaciones medias y largas el uso de un único lenguaje de programación aumenta el rendimiento en cuanto a tiempo y memoria
Memoria	No libera la memoria reservada para los paquetes hasta que la simulación termina.	Permite el uso de “dummy bytes” en los paquetes reduciendo la memoria necesaria.  La pila de protocolos completa no es necesaria en todos los nodos.
Módulos y clases	Debido a su longevidad, muchos grupos han contribuido a crear una extensa librería.	Únicamente unas pocas clases han sido portadas desde ns-2
Otras funcionalidades		Emulación de interfaces para poder integrar el simulador con redes reales.

Tabla B.1: Comparación ns-2 y ns-3

Su desarrollo surgió de forma paralela a Network Simulator 2 por lo que no es una continuación de este y los scripts de configuración desarrollados para esta versión no pueden ser portados a ns-3 ya que no son compatibles entre ellos. Sin embargo, gran parte de las librerías del núcleo de ns-2 han sido portadas y mejoras para ser funcionales en ns-3. En la actualidad el uso de ns-3 sigue en aumento y la comunidad continúa ampliando y manteniendo sus funcionalidades, al contrario que ns-2, cuyo mantenimiento ya fue abandonado hace un tiempo.

La principal ventaja de ns-3 es que ha sido desarrollado íntegramente en C++, lo que facilita muchísimo el aprendizaje a los nuevos usuarios y permite un mayor grado de control y acceso a los diferentes módulos y clases del simulador. El hecho de emplear un único lenguaje de programación es la gran ventaja que ns-3 presenta frente a su versión anterior, que empleaba una combinación de C++ para el núcleo y Object Tcl (oTcl) para los scripts de configuración de la simulación. Además, se ha añadido soporte para Python, añadiendo ciertas capacidades a los scripts que facilitan la interacción del usuario con las distintas clases del núcleo del simulador.

## **B.2 Configuración de la simulación**

Todo el código fuente que es lanzado en las versiones oficiales de ns-3 cuenta con una documentación de API que se ha realizado empleando la herramienta Doxygen, un generador de documentación que facilita mucho el acceso y el aprendizaje de los diferentes módulos que conforman el núcleo del simulador.

En una primera toma de contacto con el simulador lo más recomendable es seguir el tutorial [30] que se encuentra disponible en la página web oficial, donde se nos guía en el proceso de instalación, compilación y ejecución del programa. Asimismo, a partir de algunos ejemplos que incluye ya compilados ns-3, se puede ver la estructura básica que un script de configuración debe tener así como el empleo de funciones relacionadas con la depuración de código y el registro de los diferentes eventos que ocurren a lo largo de las simulaciones.

Una vez ya sepamos como ejecutar y recompilar el código del simulador toda la documentación [31] que necesitaremos para modificar el comportamiento del simulador para adaptarlo a nuestras necesidades estará en el repositorio de Doxygen que encontramos en la página web oficial. En el caso de este trabajo, aunque se van a emplear una gran cantidad de módulos para realizar las simulaciones, las clases que se van a modificar se encuentran únicamente dentro de tres módulos: “*applications*”, “*mobility*” y “*wifi*”.

Con todas estas consideraciones, las siguientes líneas pretenden ser una pequeña documentación sobre las peculiaridades y dificultades que surgen durante la configuración de un escenario de simulación como el que se ha empleado en este trabajo, es decir, un escenario híbrido con parte cableada (CSMA/CD) y parte inalámbrica (802.11g en modo infraestructura).

Antes de introducirnos en ningún modelo en concreto, es importante conocer que las clases configurables directamente por el usuario desde el script de simulación (sin necesidad de modificar la implementación de estas) cuentan con clases “Helper” que ofrecen un interfaz más amigable que agiliza la configuración y la instalación de los diferentes módulos en los nodos.

En un escenario híbrido con parte cableada y parte inalámbrica como el que se ha simulado en este trabajo nos interesa emplear los siguientes Helper:

- MobilityHelper: usando este Helper podemos incorporar fácilmente movimiento a los nodos según diferentes patrones en función de la instancia que creamos, pudiendo elegir entre movimientos aleatorios hasta simples desplazamientos en línea recta.
- CsmaHelper: permite añadir un interfaz CSMA/CD como Ethernet a los nodos y configurar en el momento de su instalación los parámetros básicos como el tamaño MTU, la dirección MAC y el tipo de trama (Ethernet II / 802.11 LLC/SNAP).
- WifiHelper: facilita la creación de interfaces Wi-Fi al permitir en una única instrucción añadirlos fácilmente a los nodos. Para su uso es necesario haber definido previamente otros Helper que correspondan a los diferentes niveles que componen un dispositivo Wi-Fi.
  - YansWifiChannelHelper
  - YansWifiPhyHelper
  - WifiMacHelper

Posteriormente se explicará en este mismo anexo más en profundidad la estructura que presenta la implementación del modelo Wi-Fi.

- InternetStackHelper: una vez creados los interfaces de nivel de enlace es necesario completar la pila de protocolos añadiendo los niveles IP y de transporte. Con este Helper se añaden a los nodos las instancias necesarias para permitir tanto el encaminamiento IP (añadiendo una dirección al interfaz) como la resolución de direcciones a través del protocolo ARP.
- BridgeHelper: clase importante que permite el intercambio de paquetes entre diferentes interfaces de diferente tipo al añadir a los nodos la capacidad de actuar como un bridge e interconectar diferentes segmentos LAN. Esta clase está basada en el estándar IEEE 802.1D.
- OnOffHelper: permite añadir una aplicación que genera tráfico con un patrón ON-OFF, pudiendo elegir tanto el protocolo de transporte bajo el que se encapsulara, la tasa de generación de paquetes, el tamaño del paquete y la duración de los periodos de ON y OFF.
- PacketSinkHelper: en complemento con el Helper anterior, este crea una instancia de una aplicación que recibe tráfico y lleva un conteo de los bytes que ha recibido a lo largo de la simulación.

Un ejemplo sencillo, omitiendo gran parte de los atributos configurables, de cómo crear un nodo con todos los niveles de la pila de protocolos, exactamente igual que los que existen en el escenario de simulación empleado en este trabajo y donde se puede ver el uso de las clases Helper, es el siguiente. El ejemplo presenta la configuración de un nodo que corresponde con un punto de acceso que tiene un interfaz Ethernet y otro Wi-Fi configurado en modo infraestructura.

```
//Declaración del contenedor de nodos
NodeContainer nodes;
nodes.Create(1);

//Instalación de la pila de protocolos
InternetStackHelper stack;
stack.Install (nodes);
//Configuración de la LAN Ethernet
CsmaHelper csma;
NetDeviceContainer csmaDev = csma.Install (nodes);

//Configuración del estándar WiFi
WifiHelper wifi = WifiHelper::Default ();
wifi.SetStandard (WIFI_PHY_STANDARD_80211g);
wifi.SetRemoteStationManager ("ns3::AarfWifiManager", "RtsCtsThreshold", UintegerValue
(200));

//Configuración del canal: pérdidas y delay
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
Ptr< YansWifiChannel > wifiChannelInstance = wifiChannel.Create ();

//Configuración del nivel físico: potencia, número de canal y
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
wifiPhy.SetChannel (wifiChannelInstance);
wifiPhy.Set("Frequency", UintegerValue (2412));
wifiPhy.Set("EnergyDetectionThreshold", DoubleValue (-96));

//Configuración del nivel MAC (posteriormente se va modificando según sea AP o STA)
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
wifiMac.SetType ("ns3::ApWifiMac",
                "Ssid", SsidValue (Ssid("WLAN")),
                "BeaconInterval", TimeValue (MilliSeconds(100)));

//Instalación del interfaz Wi-Fi en el nodo
NetDeviceContainer apDev = wifi.Install (wifiPhy, wifiMac, nodes.Get (0));

//Instalación de la capacidad de funcionar como bridge
NetDeviceContainer bridgeDev
    = bridge.Install (nodes.Get (0), NetDeviceContainer (apDev, csmaDev.Get (0));
```

```
//Asignación de dirección IP
Ipv4AddressHelper ip;
ip.SetBase ("192.168.0.0", "255.255.255.0");
Ipv4InterfaceContainer apInterface = ip.Assign (bridgeDev);

//Movilidad de los nodos (en este caso, al ser un AP que no se mueve, establecemos la posición
//y asignamos un modelo de posición estática)
Ptr<ListPositionAllocator> positionAlloc = CreateObject<ListPositionAllocator> ();
positionAlloc->Add (Vector (85,0, 85,0, 0.0));
MobilityHelper SetPositionAllocator (positionAlloc);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodes.Get (0));
```

Código B.1: Ejemplo de configuración de un nodo con interfaces CSMA/CD y CSMA/CA.

## **B.3 Modelo Wi-Fi en ns-3**

Una vez completado el script de configuración básico del escenario el siguiente paso para conseguir una simulación que cumpla los requisitos que necesitamos es conocer la implementación que los módulos tienen en el núcleo del simulador para ajustar los diferentes atributos configurables o modificar el comportamiento según nuestras necesidades. En el caso en el que nos encontramos el módulo que resulta más interesante conocer es el que implementa los diferentes estándares de la familia 802.11 [32].

En la siguiente figura se puede ver de un simple vistazo la estructura de clases que presenta el módulo implementado en el simulador del estándar 802.11.

La primera clase que no se encuentra en el diagrama porque realmente no contiene ningún algoritmo que realice funciones propias del estándar 802.11 es la clase WifiHelper, que ya se comentó anteriormente. Gracias a esta clase se pueden instalar (una vez definidos todos los niveles: canal, físico y MAC) los dispositivos Wi-Fi en los nodos de forma sencilla, pudiendo seleccionar el estándar 802.11 que queremos emplear (en la actualidad se encuentran implementados 802.11a/b/g/n, aunque no completamente ya que presentan algunas limitaciones).

Comenzando por el nivel más bajo nos encontramos con la implementación del canal radio en la clase YansWifiChannel y con la del nivel físico en YansWifiPhy. Estas dos clases contienen el código procedente del modelo desarrollado en Yet Another Network Simulator [24][33] para la capa física definida en el estándar 802.11.

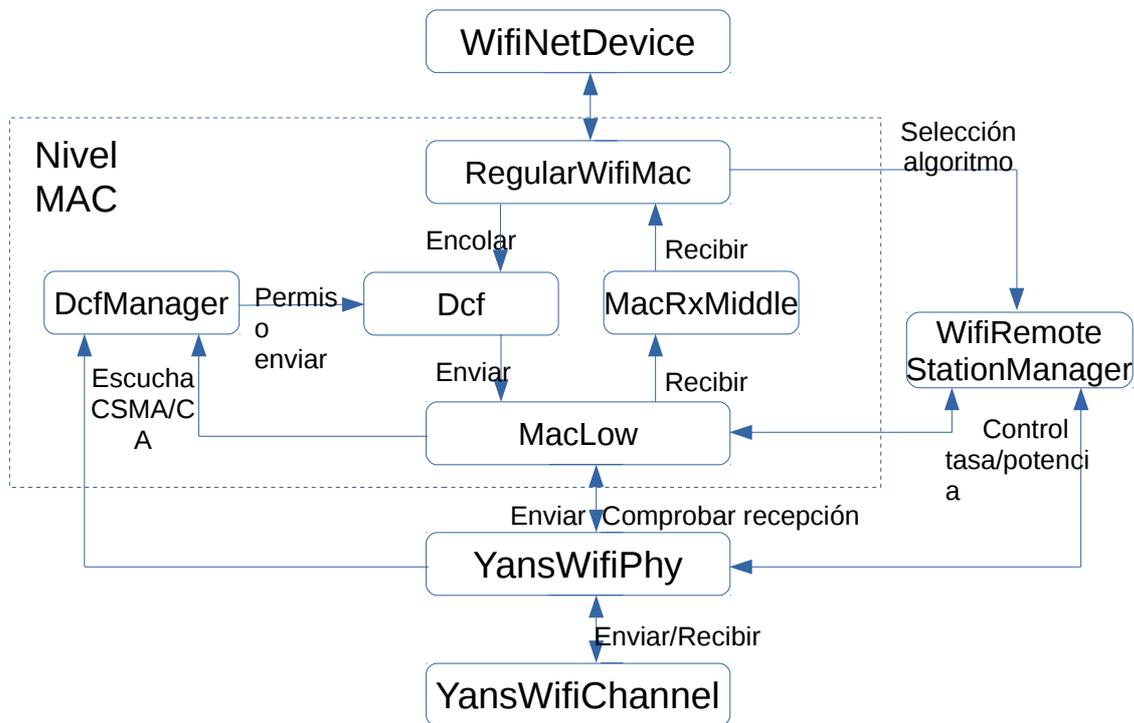


Figura B.1: Clases de ns-3 del módulo Wi-Fi

La primera clase, YansWifiChannel, es muy sencilla y es donde se realizan los cálculos relacionados con la propagación, es decir, la atenuación y el retardo que sufre la señal, pudiendo escoger en ambos casos entre varios modelos que se encuentran disponibles. Por la forma que tiene esta clase ya surge la primera limitación importante en la simulación de entornos Wi-Fi ya que no es posible la interacción entre diferentes instancias de esta clase, no permitiendo escenarios con canales solapados espectralmente.

La clase YansWifiPhy ya cuenta con una implementación más extensa, contando con un gran número de atributos configurables sin necesidad de modificar el código. La mayoría de estos atributos están relacionados con la simulación de lo que sería una tarjeta o antena Wi-Fi ya que en esta clase es donde se tienen en cuenta los umbrales de detección de energía, las ganancias de potencia en transmisión y recepción, la figura de ruido o la frecuencia de funcionamiento. En caso de trabajar con el estándar 802.11n también podríamos configurar la anchura espectral del canal radio o el número de antenas pues tiene soporte para la tecnología MIMO.

Adicionalmente, en esta clase es donde se aplica el modelo de errores a nivel de bit según el esquema de modulación (MCS) empleado, por lo que si queremos personalizar el comportamiento de alguna de estas variables, como por ejemplo el nivel de señal a ruido más interferencia, bastará con extraerlas y guardarlas en los objetos de esta clase. Los algoritmos de gestión de potencia también tienen sus efectos en esta clase.

Subiendo por la pila de protocolos nos encontramos con la implementación del nivel MAC, que a diferencia del nivel físico, se encuentra dividido en una gran cantidad de clases debido a la mayor complejidad que presenta. En la figura B.1 solo se presentan las clases principales habiendo muchas más que dependen de estas.

- MacLow: esta es la clase más baja del nivel MAC y es donde una vez el terminal ha determinado que puede transmitir, se realiza tanto el envío como la recepción de los paquetes de datos y de sus correspondientes ACK. Además, esta clase también incluye el envío y recepción de los paquetes del mecanismo de protección RTS/CTS destinado a reducir el número de colisiones y a solucionar el problema del nodo oculto y expuesto.
- DcfManager: esta clase implementa el algoritmo de acceso al medio CSMA/CA establecidos en los estándares 802.11a/b/g/n. Para indicar a la estación que puede comenzar a transmitir cuenta con dos subclases “Listener” que están continuamente monitorizando tanto el nivel físico como la clase MacLow. Así, cuando determina que puede transmitir en función de los parámetros DIFS, SIFS... y de si el mecanismo RTS/CTS está activo se le comunica a la clase Dcf que un paquete puede ser enviado.
- Dcf: esta clase abstracta se encarga de definir las funciones que gestionan las políticas de retransmisión, de la fragmentación de los paquetes y de los algoritmos relacionados con la ventana de contienda, encontrándose la implementación del código en las siguientes dos subclases.
  - DcaTxOp: implementa una cola FIFO sin ningún tipo de priorización de paquetes (ni siquiera los de gestión frente a los de datos) que no permite distinguir entre paquetes y por lo tanto no hay ninguna posibilidad de ofrecer calidad de servicio a los nodos que emplean esta opción. Cuenta con la opción de descartar los paquetes cuando han superado un tiempo umbral de estancia en la cola.
  - EdcaTxopN: implementación de la cola Enhanced Distributed Channel Access descrita en el estándar 802.11e que permite ofrecer calidad de servicio al poder distinguir los paquetes según cuatro posibles clases. Estas clases son “Voice”, “Video”, “Best-effort” y “Background”, variando los valores del tamaño de la ventana de contienda y el tiempo AIFS según indica el estándar.
- MacRxMiddle: clase sencilla que se encarga únicamente de detectar paquetes duplicados y de reconstruir los fragmentos en caso de que sea necesario.
- RegularWifiMac: es la clase más importante de todo el modelo Wi-Fi ya que engloba, controla y coordina el funcionamiento de todas las clases mencionadas del nivel MAC anteriormente. Cuenta con diferentes implementaciones según el tipo de despliegue que estemos realizando, permitiendo tanto modo infraestructura, modo adhoc y redes inalámbricas malladas.

- AdHocMac: clase sencilla que implementa el código necesario para simular el comportamiento de un nodo que se encuentra en una red inalámbrica descentralizada donde las estaciones se comunican directamente entre sí.
- MeshWifiInterfaceMac: clase que contiene el código del nivel MAC específico de nodos que se encuentran en redes inalámbricas malladas, es decir, redes Wi-Fi donde se encuentran mezclas topologías ad hoc con topologías en modo infraestructura.
- OcbWifiMac: implementación del nivel MAC definida en el estándar 802.11p-2010 destinado a añadir acceso inalámbrico a automóviles para permitir el intercambio de datos entre ellos y entre infraestructura de carretera.
- ApWifiMac: clase que implementa el comportamiento de los puntos de acceso (AP) en despliegue configurado en modo infraestructura. Las funciones propias de esta clase son la generación de las tramas Beacon (permitiendo modificar el tiempo de generación entre ellos además de simular un posible jitter) y los procedimientos relacionados con la asociación de STAs, es decir, la respuesta tanto a las tramas “Probe Request” como a los mensajes de asociación, pudiendo ser esta última negativa si las tasas soportadas por la STA no corresponden con las que funciona el AP.

Debido a como está implementado el acceso a la cola definida en la clase Dcf puede surgir un problema al no priorizar los paquetes de gestión sobre los de datos, lo que puede provocar que en momentos donde hay congestión en el downlink los procedimientos de asociación se vean ralentizados, degradando las prestaciones de las STA. Únicamente los paquetes Beacon no sufren este problema ya que cuentan con una cola propia que tiene prioridad en el envío de los datos frente a los paquetes de datos.

- StaWifiMac: esta clase complementa a la anterior en una topología en modo infraestructura, teniendo que instalarla en los nodos que se comporten como STAs. Así, esta clase contiene los algoritmos relacionados con la recepción de tramas Beacon y la generación de los paquetes “Association Request” y “Probe Request”, pudiendo elegir si la STA es activa o pasiva a la hora de asociarse.

Sin embargo la implementación de esta clase dista mucho de simular completamente el funcionamiento que una STA tiene en la realidad ya que presenta algunas deficiencias que comprometen y limitan su comportamiento. El principal inconveniente es la gestión que hace en el proceso de asociación y reasociación ya que la elección del punto de acceso no se realiza por ninguna métrica como podría ser el nivel de potencia recibida, sino que en cuanto se recibe un Beacon válido (la STA se configuran para que solo se conecten a los AP que tienen un BSSID concreto). En la figura B.2 se muestra la máquina de estados de la STA según el momento del proceso de asociación en el que se encuentre.

Otra limitación importante y que está relacionado con la limitación ya comentada anteriormente de la clase YansWifiChannel es que cada instancia de la clase StaWifiMac solo puede estar escuchando en un único canal a la vez, no pudiendo recibir Beacons de varios canales de forma simultánea.

Además, el problema anteriormente comentado de la no priorización de los paquetes de gestión también se reproduce en este caso, agravando todavía más los tiempos necesarios para las asociaciones con los AP.

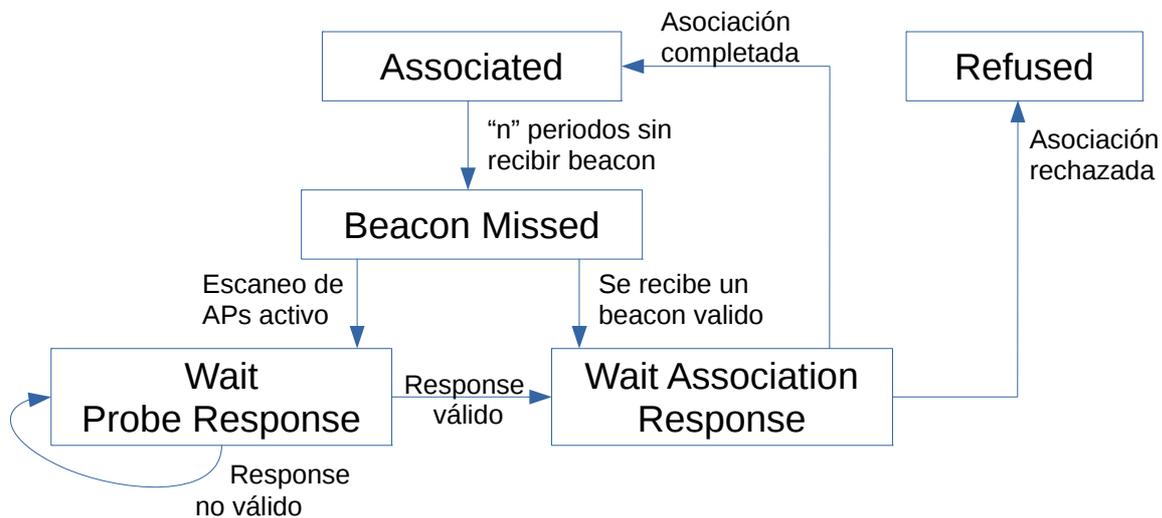


Figura B.2: Máquina de estados de las estaciones móviles en modo infraestructura

En medio de la capa física y el nivel MAC se encuentra la clase `WifiRemoteStationManager` que es muy importante configurar ya que tiene un gran impacto en los resultados de las simulaciones. La función de esta clase consiste en realizar los algoritmos de selección de potencia en el nivel físico y de tasa en el nivel MAC según los parámetros configurados (DIFS, SIFS, CTS...), de los niveles de potencia recibidos y de si las transmisiones se están completando correctamente. Existen varias implementaciones de estos algoritmos, pero las más usadas y documentadas son las siguientes:

- [ArfWifiManager](#) [6]: algoritmo que pretende maximizar el throughput de cada STA aumentando la tasa con cada 10 paquetes seguidos transmitidos satisfactoriamente y disminuyéndola cuando ocurren retransmisiones. El principal problema es que no es capaz de adaptarse a un canal con unas condiciones muy cambiantes.
- [AarfWifiManager](#) [7]: versión mejorada del anterior algoritmo donde los umbrales para aumentar la tasa varían de forma dinámica para adaptarse a las condiciones del escenario.
- [AarfcdWifiManager](#) [8]: algoritmo muy parecido al anterior pero que añade un nuevo parámetro al modificar dinámicamente la presencia o no del mecanismo RTS/CTS en función del número de retransmisiones y paquetes transmitidos de forma satisfactoria.
- [MinstrelWifiManager](#) [34]: algoritmo de control de la tasa desarrollado inicialmente en el controlador MadWifi para sistemas operativos Linux. Su implementación consiste en la elaboración de una tabla con las posibilidades de que la transmisión a una determinada tasa

sea satisfactoria, no habiendo una relación directa entre el nivel de potencia y la tasa de transmisión.

- CaraWifiManager [9]: algoritmo basado en ARF pero que mejora su funcionamiento en situaciones donde hay colisiones, empleando el mecanismo RTS/CTS para determinar si las transmisiones fallidas se han debido al uso de una modulación demasiado elevada o por una colisión.

Por último, el nivel superior del modelo Wi-Fi es WifiNetDevice, que incluye todas las clases comentadas anteriormente, desde el canal físico hasta la capa superior del nivel MAC, ofreciendo un interfaz único hacia los niveles superiores. Esta clase se podría considerar como la implementación de la tarjeta de red Wi-Fi ya que es desde donde se coordina todos los elementos definidos en el estándar 802.11 además de ser el punto donde se asigna la dirección MAC o desde donde se genera la resolución de direcciones IP a través del protocolo ARP.



# Anexo C. Soporte para despliegues 802.11 multicanal sin solape espectral en ns-3.

## C.1 Limitaciones de ns-3

Inicialmente, cuando la tecnología 802.11 se encontraba todavía poco extendida, los despliegues consistían únicamente en un único punto de acceso trabajando en un único canal frecuencial pero en la actualidad es habitual encontrarse con un espectro Wi-Fi sobreutilizado y saturado, por lo que se emplean todos los canales posibles. Así, para que los resultados de las simulaciones sean lo más reales posibles, es necesario que los programas soporten el uso de varios canales de forma simultánea.

En el caso de Network Simulator 3, tal como ya se planteó en el anexo B, cuenta con varias deficiencias en su implementación actual que imponen ciertas limitaciones en los posibles escenarios a simular. La primera limitación se encuentra en la clase YansWifiChannel y consiste en la imposibilidad de que varios canales Wi-Fi sufran interferencias entre ellos. De esta forma, sin rediseñar todo el sistema de interferencias de ns-3, algo realmente costoso y que no se va a realizar en este trabajo, la única opción es trabajar con frecuencia

Nº	Frecuencia central [MHz]	Limite inferior [MHz]	Limite superior [MHz]
1	2412	2401	2423
2	2417	2406	2428
3	2422	2411	2433
4	2427	2416	2438
5	2432	2421	2443
6	2437	2426	2448
7	2442	2431	2453
8	2447	2436	2458
9	2452	2441	2463
10	2457	2446	2468
11	2462	2451	2473
12	2467	2456	2478
13	2472	2461	2483

Tabla C.1: Utilización de la banda de 2,4GHz para Wi-Fi en Europa

En caso de emplear la banda de 5GHz de 802.11a, esta limitación en el diseño del canal no tienen ningún efecto ya que los canales no presenta un solape espectral entre ellos, sin embargo, si simulamos la banda de 2.4GHz con cualquiera de los estándares 802.11b/g/n sí que deberemos tener en cuenta ya que como se ve tanto en la tabla C.1 y en la figura C.1 existe un gran solape debido a que el ancho de banda es de 22MHz y la separación entre canales es de solo 5MHz por lo que en consecuencia hay una gran interferencia entre canales.

La segunda limitación está en la implementación del nivel MAC de los terminales móviles, y más en concreto en la clase StaWifiMac, donde las STAs no pueden realizar un escaneo de diferentes canales de forma simultánea para escuchar los Beacon o enviar Probe Request para asociarse a las APs que vean más conveniente hacerlo. Estas limitaciones son un tema poco recurrente en la bibliografía y en muchas simulaciones que emplean ns-3 no se tienen en cuenta, únicamente en [35] se discute de la necesidad de añadir el soporte multicanal a ns-3 para aumentar el throughput global del sistema al multiplicar el espectro útil. Este problema, al contrario que el anterior, sí que puede ser resuelto sin necesidad de desarrollar desde cero todo el modelo Wi-Fi, pudiendo conseguir unos buenos resultados modificando únicamente ciertos comportamientos del nivel MAC.

La solución propuesta aquí para emular de forma sencilla la posibilidad de escaneo de varias frecuencias sin necesidad de modificar de forma extensa el código consiste en la creación en las STAs de tantas capas MAC con sus respectivos niveles físicos, como canales queramos escanear. En este trabajo, basta con considerar 3 capas MAC/Física por los motivos que se exponen a continuación.

En primer lugar debemos ser conscientes de que cualquier solución para añadir soporte para varios canales de forma simultánea está condicionada porque estos no tengan solape espectral y en la figura C.1 se puede observar el solape que existe entre ellos. Así, los canales elegidos para la simulación solo podrán ser tres y en este caso se ha optado por escoger los que se encuentran en color azul, es decir, los canales 1, 6 y 11.

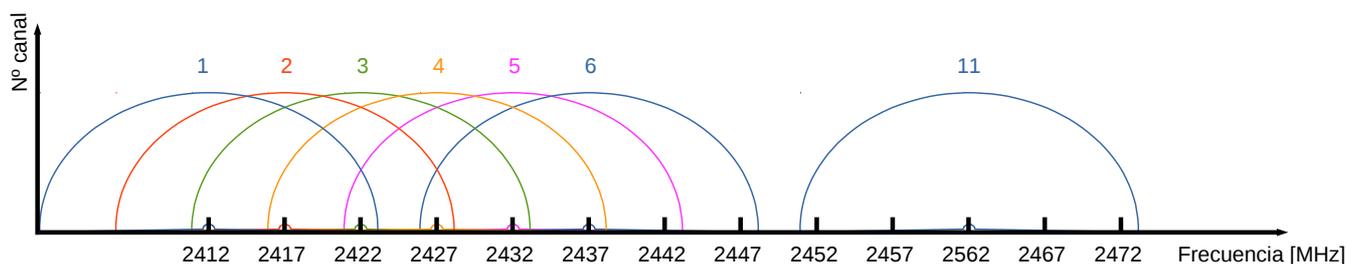


Figura C.1: Solapes entre canales Wi-Fi.

## C.2 Implementación en ns-3

La implementación de la solución para la emulación del escaneo multicanal se debe realizar tanto en el script de configuración como en algunas clases del modelo Wi-Fi (ver anexo B para conocer más detalles). En el caso del script de configuración del escenario el código es el siguiente:

```
// Es necesario crear tantas instancias de canal y de sus correspondientes niveles físicos
// como canales no solapados queramos tener.
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
Ptr< YansWifiChannel > wifiChannelInstance1 = wifiChannel.Create ();

Ptr< YansWifiChannel > wifiChannelInstance2 = wifiChannel.Create ();
Ptr< YansWifiChannel > wifiChannelInstance3 = wifiChannel.Create ();

YansWifiPhyHelper wifiPhy1 = YansWifiPhyHelper::Default ();
wifiPhy1.SetChannel (wifiChannelInstance1);
YansWifiPhyHelper wifiPhy2 = YansWifiPhyHelper::Default ();
wifiPhy2.SetChannel (wifiChannelInstance2);
YansWifiPhyHelper wifiPhy3 = YansWifiPhyHelper::Default ();
wifiPhy3.SetChannel (wifiChannelInstance3);
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
NodeContainer sta;
NetDeviceContainer staDev;
NetDeviceContainer dev;
Ipv4InterfaceContainer staIf;

//Creación del canal 1
wifiMac.SetType ("ns3::StaWifiMac",
                "Ssid", SsidValue (Ssid("WLAN")),
                "ActiveProbing", BooleanValue (false),
                "IdNode", UintegerValue ( sta.Get(j)->GetId() ),
                "WifiChannel", UintegerValue (1));
staDev.Add(wifi.Install (wifiPhy1, wifiMac, sta.Get(j)));
staIf = ip.Assign (staDev.Get(j));

//Creación y configuración del canal 2
wifiMac.SetType ("ns3::StaWifiMac",
                "Ssid", SsidValue (Ssid("WLAN")),
                "ActiveProbing", BooleanValue (false),
                "IdNode", UintegerValue ( sta.Get(j)->GetId() ),
                "WifiChannel", UintegerValue (2));
dev.Add(wifi.Install (wifiPhy2, wifiMac, sta.Get(j)));

// Con estos comandos lo que se hace es asignar una @MAC y una @IP única a los interfaces
```

```
dev.Get(0)->SetAddress(staDev.Get(j)->GetAddress());
Ptr<NetDevice> device = dev.Get(0);
Ptr<Node> node = sta.Get(j);
Ptr<Ipv4> ipv4 = node->GetObject<Ipv4> ();
int32_t interface = ipv4->GetInterfaceForDevice (device);
if (interface == -1)
    interface = ipv4->AddInterface (device);

Ipv4InterfaceAddress ipv4Addr = Ipv4InterfaceAddress (staIf.GetAddress(0), "255.255.255.0");
ipv4->AddAddress (interface, ipv4Addr);
ipv4->SetMetric (interface, 1);
ipv4->SetUp (interface);

//Creación y configuración del canal 3
wifiMac.SetType ("ns3::StaWifiMac",
                "Ssid", SsidValue (Ssid("WLAN")),
                "ActiveProbing", BooleanValue (false),
                "IdNode", UIntegerValue ( sta.Get(j)->GetId() ),
                "WifiChannel", UIntegerValue (3));
dev.Add(wifi.Install (wifiPhy3, wifiMac, sta.Get(j)));

// Con estos comandos lo que se hace es asignar una @MAC y una @IP única a los interfaces
dev.Get(1)->SetAddress(staDev.Get(j)->GetAddress());
device = dev.Get(1);
interface = ipv4->GetInterfaceForDevice (device);
if (interface == -1)
    interface = ipv4->AddInterface (device);

ipv4->AddAddress (interface, ipv4Addr);
ipv4->SetMetric (interface, 1);
ipv4->SetUp (interface);
```

Código C.1: Script de configuración del escenario con soporte multicanal

Todo el código anterior tiene el único propósito de generar tres instancias diferentes, una para cada canal, de la clase `WifiNetDevice` que tengan la misma dirección MAC e IP para hacer que su presencia sea transparente para el resto de nodos y vean que hay un único interfaz en esa STA. Al hacer esto se deberá tener especial cuidado de que nunca dos o más enlaces estén activos al mismo tiempo ya que se generaría un conflicto en la resolución ARP al responder varios interfaces a través de diferentes APs de forma simultánea.

Además, si queremos conseguir una itinerancia real de las STA donde todos los paquetes encolados no se pierdan y las conexiones ya establecidas sigan funcionando correctamente después de realizar un traspaso se deben añadir las siguientes líneas cuando creamos las aplicaciones en las

STAs para almacenar en una variable los socket de las aplicaciones donde se va a transmitir el tráfico generado en el mismo nodo (ver código C.2).

---

```
Ptr<Socket> socket = Socket::CreateSocket(staNodes[i].Get(j), UdpSocketFactory::GetTypeId ());
Ptr<OnOffApplication> app = DynamicCast<OnOffApplication> (node->GetApplication (k));
  app->SetSocket(socket);

StaWifiMac::sockets[node->GetId()] = socket;
```

Código C.2: Configuración de las aplicaciones en el escenario con soporte multicanal

Una vez que tenemos declarados todos los interfaces con sus respectivos canales la clase que más hay que modificar es `StaWifiMac` ya que es donde se deben gestionar la captura de los Beacon y tomar la decisión de que canal va a emplear la STA para transmitir. Las modificaciones están encaminadas a colocar la cola de transmisión en el interfaz activo y a asociar el socket al interfaz de red por el que queremos que se envíe el tráfico (en ns-3 las aplicaciones no se asocian dinámicamente, solo al comienzo, por lo que si el interfaz activo se cae o se desactiva el tráfico no se redirige hacia otro que este activo).

Para que la nueva clase `StaWifiMac` funcione es necesario introducir ciertas variables estáticas a la clase que permitan a las diferentes instancias de un mismo nodo comunicarse entre ellas. Además, es necesario introducir algunos datos adicionales como las direcciones MAC de todos los puntos de acceso y del canal en el que están operando, para así poder identificar correctamente el origen de todos los Beacon.

En cuanto al código necesario para la gestión de las colas, sockets y activación o desactivación de los interfaces, sin tener en cuenta la toma de datos o la inicialización de las variables para no extender demasiado la muestra de código, es el siguiente:

---

```
//Variables estáticas declaradas en StaWifiMac.h
static std::vector<Ptr<Socket> > sockets;
static std::vector <std::vector<Ptr<WifiMacQueue> > > macQueues;

//Instrucciones que actualizan la cola, el socket y el estado del interfaz en el momento donde se
//determina la necesidad de hacer un handover
macQueues[m_idNode][apChannels[apId]-1]
  ->SetPacketQueue(macQueues[m_idNode][m_channel-1]->GetPacketQueue());
macQueues[m_idNode][apChannels[apId]-1]
  ->SetSize(macQueues[m_idNode][m_channel-1]->GetSize());
macQueues[m_idNode][m_channel-1]->Flush();

sockets[m_idNode]
  -> BindToNetDevice(WifiHelper::netDevices[m_idNode][apChannels[apId]-1]);
```

```
SetState (BEACON_MISSED);  
m_linkDown();  
IsReady();
```

Código C.2: Gestión de la cola y los sockets en la clase StaWifiMac.

Una breve explicación del anterior código, que se puede dividir en tres partes según la operación que estemos realizando, es la siguiente:

1. Las primeras instrucciones están encaminadas a indicar a la instancia DcaTxOp del nuevo interfaz activo el tamaño que tiene la cola y cuál es la cola que debe emplear y a la del interfaz viejo eliminar cualquier paquete o tamaño residual con la función Flush().
2. La segunda parte es solo una instrucción para solucionar el problema ya comentado de los sockets que se asocian a un único interfaz al comenzar la aplicación.
3. Las tres últimas órdenes tienen la función de apagar las posibles respuestas ARP o ICMP del interfaz y de permanecer a la espera de que se determine que vuelve a activarse para asociarse con un nuevo AP.

Con todos estos cambios, la máquina de estados original, explicada en el anexo B, de las STAs sufre algunos cambios, a la vez que introduce un componente ajeno a la interacción con la llegada de los Beacon y se elimina el “watchdog” que tras “n” periodos sin recibir un Beacon de nuestro AP determinaba cuando una estación tiene que pasar a un estado donde no está asociada. Ahora, si durante el periodo de escucha de los Beacon, determinado por el atributo AlgorithmInterval, no se recibe ningún Beacon del AP al que estamos asociados, se determina que se ha perdido la conexión y se realiza un handover a otro AP en caso de haber algún candidato.

Con todas estas modificaciones lo que se pretende es modificar el comportamiento del modelo Wi-Fi de ns-3 del modo infraestructura en las STAs para que tenga un único interfaz tanto en el nivel inferior (MAC) como en el nivel superior (IP). En la figura C.2 se puede ver el modelo resultante con los cambios introducidos.

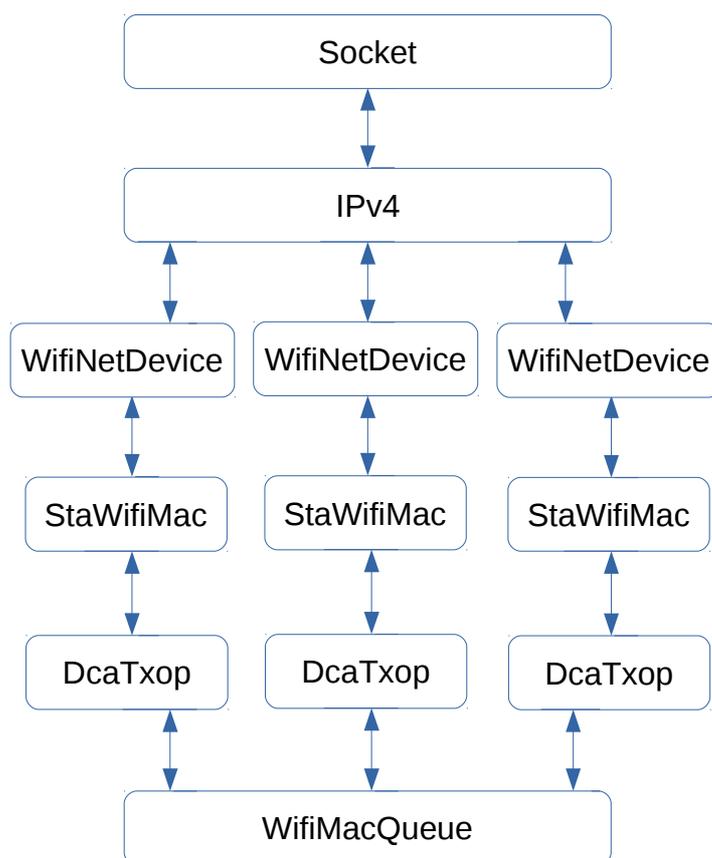


Figura C.2: Modelo de clases con soporte multicanal

Para simular un comportamiento más realista de las STAs también se ha modificado el proceso de encolar un paquete en la clase WifiMacQueue, dando prioridad a los paquetes de gestión (Association/Probe Request/Response) frente a los paquetes de datos. En caso de no realizar este cambio los procesos de handover podrían ralentizarse en caso de que las colas del nivel MAC se encuentren saturadas. La probabilidad de que ocurra este problema es muy alta ya que las STAs suelen reasociarse, según el criterio de asociación empleado, cuando las condiciones del canal se han degradado mucho y por lo tanto se ha transmitido con tasas bajas (típicamente 1Mbps) en los instantes previos al handover, por lo que en caso de retrasar el proceso de handover, todo el tiempo que se demore el paquete Association Request será también otro tiempo en el que el buffer se siga llenando, haciendo que el problema se haga todavía mayor.

Todos estos cambios para soportar múltiples canales de forma simultánea no tendrían demasiado sentido sin introducir una gestión inteligente en la forma de gestionar el roaming ya que como ya se ha indicado anteriormente, la implementación por defecto no soporta reasociaciones de las STAs puesto que hasta que no pierde la asociación con su actual AP no se plantea asociarse con otro. Además, estas asociaciones no se realizan por ningún criterio sino que la STA se asocia con el primer AP del que recibe un Beacon válido. En el siguiente anexo se explican estos cambios que modifican por completo la forma en la que se comportan las STAs en ns-3.



# Anexo D. Implementación de algoritmos de asociación en redes en modo infraestructura en ns-3.

## D.1 Métricas

Continuando con las modificaciones necesarias para conseguir el funcionamiento necesario para evaluar las distintas estrategias a la hora de asociar las STAs a los distintos APs para balancear la carga en escenarios tanto estáticos como cambiantes el último paso es implementar la selección de las métricas a emplear, la toma de estas variables en las clases pertinentes y la implementación de los algoritmos de decisión.

En cuanto a las posibles variables a emplear en la decisión de asociarse a un punto de acceso u a otro se han escogido las siguientes seis:

- Número de estaciones asociadas
  - Esta métrica es muy simple, únicamente consiste en revisar el número de STAs asociadas en el momento en el que se recibe el Beacon.
  - La implementación contará con un margen de histéresis en la comparación que deberá ser siempre superior o igual a 2 para evitar que una estación esté constantemente desequilibrando el balanceo. Este margen podrá ser establecido según las necesidades del escenario pudiéndolo cambiar en función del número de STAs presentes en el escenario.
  - Toda la gestión de esta variable se puede realizar fácilmente desde la clase StaWifiMac ya que ahí es donde se completan las asociaciones, por lo que se puede llevar sin demasiadas complicaciones el conteo del número de usuarios vinculados a cada AP.
  
- Distancia
  - En primer lugar hay que destacar que esta variable no se puede emplear en la realidad con la misma precisión ya que es imposible conocer la distancia real entre el AP y la STA. Además, a priori, no tendría demasiada utilidad puesto que generalmente no existe visión directamente entre ambos. Sin embargo el motivo de que se haya incluido es que permitirá ver los efectos del shadowing en las métricas de RSSI y SINR puesto que la distancia en un entorno de propagación sin shadowing es equivalente a usar la RSSI ya que solo hay que tener en cuenta las pérdidas introducidas por el modelo de propagación.
  - La implementación contará con un margen de histéresis en la comparación de distancias
  - Esta variable es de fácil acceso ya que se conocen ambos nodos y se pueden comparar sus posiciones a través de los respectivos MobilityModel asociados a los nodos y con la función GetDistanceFrom() de esa misma clase.

- RSSI (Received Signal Strength Indicator)
  - La potencia con la que se reciben los Beacon desde los puntos de acceso suele ser la métrica que más se emplea en la realidad. El empleo de esta métrica está directamente relacionado con la máxima tasa de transmisión que podemos conseguir cuando la estación este transmitiendo ya que el nivel de potencia y el margen de obtenido respecto al nivel de ruido (asumiendo la ausencia de interferencia) determinan el MCS que se puede emplear.
  - La implementación se realizará aplicando un margen de histéresis en la comparación para evitar el efecto “ping-pong”, debido al movimiento de las STAs o a los efectos del shadowing, que provocaría un exceso de reasociaciones con la consiguiente pérdida de prestaciones.
  - En una implementación serializada y completamente escalable esta variable se introduciría en la clase YansWifiChannel, que es donde se calculan las potencias con las que se reciben los paquetes, a través del sistema de “tags” que incluye ns-3 y que constituye una de las mejoras respecto a su antecesor ns-2. Finalmente se extraería en la clase StaWifiMac que es donde se realizan los procesos de asociación. Sin embargo, debido a la dificultad que implica el uso de tags las potencias de los Beacon se han guardado en una variable estática de YansWifiChannel que es un vector donde se guardan las potencias de los últimos Beacon recibidos por cada STA.
  
- SINR (Signal to Interference plus Noise Ratio)
  - Al igual que el criterio anterior, la relación señal a ruido más interferencia también es un criterio que se puede implementar en una topología ya desplegada con una simple actualización del software de las STAs.
  - La implementación será similar a la de RSSI ya que los efectos a evitar son los mismos, incluyéndose la influencia de la interferencia cocanal.
  - En este caso, esta variable se encuentra en la clase YansWifiPhy ya que es el propio nodo receptor el que una vez recibido el paquete invoca a la clase InterferenceHelper para obtener la interferencia existente durante la recepción del paquete y tras sumarle el ruido se obtiene tanto la SINR como el PER. Al igual que con la RSSI, se guarda en un vector estático para poder acceder desde la clase StaWifiMac.
  
- Carga del canal
  - Esta métrica consiste en que a la vez que cada punto de acceso monitoriza el canal, almacene en una variable el tiempo que detecta en el que se está transmitiendo un paquete (ya sea el mismo o cualquier otro dispositivo que emplee ese mismo canal). Un elemento a tener en cuenta a la hora de emplear este criterio es que la carga nunca podrá ser del 100% ya que los intervalos de espera DIFS, SIFS, PIFS y AIFS no se pueden contabilizar como carga ya que el canal realmente no está ocupado, aunque estos intervalos suponen una de las mayores fuentes de ocupación del canal debido a la estructura que se define en CSMA/CA.
  - La implementación de esta estadística de decisión volverá a contar con un umbral de decisión en forma de cociente entre la carga del actual y la carga del objetivo a la hora de realizar un handover. Sin embargo, al contrario que en las métricas anteriores, habrá un segundo umbral ya que en caso de que el canal se encuentre poco ocupado no interesara cambiar de AP puesto que la mejora no sería significativa.

- En este caso conseguir esta métrica es algo más difícil que en todos los casos anteriores ya que no se encuentra disponible directamente en una variable que se pueda guardar para consultarse posteriormente, sino que lo que se debe hacer es capturar los eventos (llamados Callbacks en ns-3) generados por la clase YansWifiPhy en los APs cada vez que detectan que el canal cambia su estado. Finalmente, el tiempo de ocupación se calculará cada medio segundo y se almacenará en la clase StaWifiMac para poder acceder a él fácilmente todas las veces que se considere necesario.
- Tasa cursada
  - Métrica muy semejante a la anterior pero en lugar de la ocupación del canal en términos de % de tiempo la variable representará el tráfico procesado por el punto de acceso durante los últimos segundos.
  - La implementación de esta métrica será exactamente igual que en el caso de la carga del canal físico, aplicando dos umbrales de decisión que deberán cumplirse para que se apruebe el handover, un umbral mínimo en la tasa a partir de la cual se podrá realizar el handover y otro que será un cociente entre la del AP actual y el objetivo.
  - En este caso, al igual que en el caso anterior, tampoco hay ninguna variable que indique directamente la tasa pero su obtención será mucho más sencilla ya que bastará con llevar un conteo de los bytes de datos que se están transmitiendo e ir haciendo un promedio cada cierto tiempo que se guardará en un vector estático para poder ser un dato accesible desde la clase StaWifiMac. Todo esto se realiza en la clase ApWifiMac, que es la clase por donde pasa todo el tráfico que reciben o envían los APs.

Adicionalmente existirían otras posibles métricas que se han descartado emplear por diferentes motivos:

- Mayor tasa de pico

Esta consistiría en escoger por ejemplo, el punto de acceso que ofrezca la posibilidad de obtener una mayor tasa de pico (por ejemplo 54Mbps de una estación con soporte para 802.11g frente a otra con 11Mbps con 802.11b). Dentro de un mismo estándar también se podrían hacer distinción ya que no es obligatorio soportar todas las tasas. El motivo para no implementar esta opción es que el escenario que se ha escogido para simular los APs no va a tener ninguna diferencia en cuanto a las tasas disponibles.
- Retrocompatibilidad con 802.11b desactivada

Este factor de decisión podría pasar desapercibido pero el impacto que tiene sobre el rendimiento real es enorme y tenerlo en cuenta a la hora de realizar el balanceo de la carga puede ser más importante que cualquier otro criterio. Así, se diferenciarían los APs que funcionan bajo 802.11b/g/n con preámbulo largo de los que funcionan con 802.11g/n con un tiempo de slot corto y por lo tanto no tienen compatibilidad con los primeros. El motivo para no implementarlo es que el simulador ns-3 no permite que dos tiempos de slots convivan de forma simultánea en una simulación.
- Elección de diferentes bandas frecuenciales

En la actualidad es normal encontrarse con redes con puntos de acceso con soporte para las dos bandas (2,4GHz y 5GHz) de forma simultánea, a la vez que el soporte para la banda de

5GHz en los terminales móviles está aumentando exponencialmente en los últimos años. La necesidad de gestionar bandas con diferentes condiciones de propagación y diferente grado de solape supone un reto ya que su correcta gestión podría multiplicar el throughput global del sistema. El motivo para no implementar esta opción es que aumentaría todavía más la complejidad en la gestión de los diferentes interfaces puesto que habría todavía más y se juntarían varios dentro de cada banda, aunque es una alternativa muy interesante donde poder seguir estudiando el impacto de los algoritmos de asociación.

## D.2 Implementación

Por cómo están guardadas todas las métricas resulta muy fácil ampliar el funcionamiento de la clase StaWifiMac (modela el comportamiento de las STAs en un despliegue en modo infraestructura) si queremos añadir un algoritmo de decisión mixta que por ejemplo emplee varias métricas ponderadas con diferentes pesos.

Las variables estáticas (compartidas por todas las instancias de StaWifiMac) añadidas a la clase son las que se muestran en el código D.1 y se pueden dividir en tres partes: la primera son los vectores que están destinadas a almacenar la información relacionada con los puntos de acceso (Id del nodo, SSID y canal en el que está transmitiendo) y que deberá ser introducida desde el script de simulación antes de arrancar la simulación (a excepción del número de usuarios asociados). En segundo lugar están los vectores donde se almacenaran las distintas métricas cada vez que se reciba un mensaje Beacon o una respuesta a un Probe. Por último se encuentran las variables que permitirán la coexistencia de diferentes interfaces en un mismo nodo y que en última instancia aseguran que el funcionamiento de la solución multicanal propuesta funcione como si se tratara de un único interfaz.

---

```
//Variables de consulta de información de los APs
static std::vector <uint32_t> apIds;
static std::vector <Mac48Address> apAddresses;
static std::vector <uint32_t> apChannels;
static std::vector <uint32_t> usersNumbers;

//Variables para almacenar la información una vez recibidos los Beacon o Probe Response
static std::vector<std::vector<uint32_t> > recvApIds;
static std::vector<std::vector<double> > recvDistances;
static std::vector<std::vector<uint32_t> > recvUsersNumbers;
static std::vector<std::vector<double> > recvApLoads;
static std::vector<std::vector<double> > recvApRates;

//Variables para permitir el funcionamiento multicanal
static std::vector <Ptr<Socket> > sockets;
static std::vector <std::vector<Ptr<WifiMacQueue> > > macQueues;
static std::vector <Mac48Address> lastMacBssid;
```

---

```
static std::vector <uint16_t> lastIdBssid;
```

Código D.1: Variables necesarias para los algoritmos de decisión en la clase StaWifiMac.

Entre las funciones implementadas se pueden diferenciar tres grupos según la función que están realizando. El primer grupo es directamente la implementación de los algoritmos de decisión y que se va a explicar posteriormente la estructura que presentan con un ejemplo. El segundo grupo corresponde con las funciones que se emplean para la adquisición y la gestión de las distintas métricas. Finalmente, IsReady() es la función donde los interfaces físicos inactivos están esperando la señal del interfaz activo de que el AP objetivo está en su canal para comenzar a funcionar.

```
// Funciones que implementan los distintos algoritmos de decisión
void decisionNUsers ();
void decisionDistance ();
void decisionRSSI ();
void decisionSINR ();
void decisionLoad ();
void decisionRate ();

// Funciones destinadas a la adquisición y gestión de las métricas
void SaveBeaconMetrics(Mac48Address ssid);
void updateAssociations (Mac48Address lastBssid, Mac48Address newBssid);
uint32_t checkAssoNumber (Mac48Address newBssid);
double calculateDistance (Mac48Address newBssid);
double checkLoadAp (Mac48Address newBssid);
    double checkRateAp (Mac48Address newBssid);
uint32_t checkApId (Mac48Address newBssid);

// Funcion donde esperan los interfaces durante su estado "Inactive"
void IsReady();
```

Código D.2: Variables necesarias para los algoritmos de decisión en la clase StaWifiMac.

La recogida de las variables se realiza dentro de la función ya existente Receive() tanto si se trata de un paquete Beacon como un Probe Response.

El código D.3 muestra un ejemplo de la implementación de los algoritmos de decisión, en este caso se trata del RSSI, pero es extrapolable a cualquier otra métrica ya que todos se encuentran estructurados de la misma forma:

- En primer lugar lo que se hace es buscar la métrica tanto de nuestro actual AP como del mejor AP objetivo.

- En función de lo obtenido en la anterior búsqueda se pueden realizar dos cosas:
    - Si el AP objetivo transmite en un canal que no se corresponde con el canal del interfaz activo y se cumplen los umbrales requeridos para realizar el handover se prepara el nuevo interfaz para que entre en funcionamiento (se asocian la cola y el socket con el nuevo interfaz) y se comunica al otro interfaz que debe entrar en funcionamiento.
    - Si no se han recibido ningún Beacon se deshabilita el interfaz y se actualiza el número de estaciones asociadas.
    - Si el AP objetivo transmite en un canal que coincide con el canal del interfaz activo y se cumplen los umbrales se envía un Request a este punto de acceso.
  - Por último se borran todas las métricas leídas durante ese intervalo y se programa dentro de “AlgorithmInterval” milisegundos una nueva ejecución del algoritmo.
- 

```
void StaWifiMac::decisionRSSI (){
    double umbral = m_rssiUmbral;
    double actual = -1000;
    double best = -1000;
    int32_t apId = -1;

    // Parte donde se analizan las metricas para decidir cual es el mejor candidato
    for (uint32_t i = 0; i < YansWifiPhy::beaconRSSI[m_idNode].size(); ++i){
        if (YansWifiPhy::beaconRSSI[m_idNode][i] > best){
            apId = i;
            best = YansWifiPhy::beaconRSSI[m_idNode][i];
        }
        if (i == lastIdBssid[m_idNode]){
            actual = YansWifiPhy::beaconRSSI[m_idNode][i];
            if (actual < -96) //Esta linea es para evitar beacons por debajo de la sensibilidad = -96dBm
                actual = -1000;
        }
    }
    // En esta parte se mira si el canal del mejor AP es el mismo que el de nuestro interfaz y en caso
    //contrario se cambian los parámetros necesarios para que el interfaz que tenga ese canal se asocie.

    //Aqui se decide si hace falta cambiar de canal y de interfaz
    if (apChannels[apId] != m_channel && (actual < (best-umbral) )){ /
        isActive[m_idNode][m_channel-1] = false;
        isActive[m_idNode][apChannels[apId]-1] = true;

        macQueues[m_idNode][apChannels[apId]-1]
            ->SetPacketQueue(macQueues[m_idNode][m_channel-1]->GetPacketQueue());
        macQueues[m_idNode][apChannels[apId]-1]
            ->SetSize(macQueues[m_idNode][m_channel-1]->GetSize());
        macQueues[m_idNode][m_channel-1]->Flush();
        sockets[m_idNode]
            ->BindToNetDevice(WifiHelper::netDevices[m_idNode][apChannels[apId]-1]);
    }
```

```
SetState (DISASSOCIATED);
m_linkDown();
IsReady();
}

else{ //AQUI SE ENTRA SI SE DETERMINA QUE NO HAY CAMBIO DE CANAL

// líneas que sustituyen el comportamiento del watchdog de beacons
if (recvApIds[m_idNode].size() == 0 && (m_state == ASSOCIATED)){
    SetState (DISASSOCIATED);
    m_linkDown ();
    updateAssociations(lastMacBssid[m_idNode], Mac48Address("ff:ff:ff:ff:ff:ff"));
    lastIdBssid[m_idNode] = 1000;
}

// líneas para mandar un Association Request
if (((lastMacBssid[m_idNode] != apAddresses[apId]) || m_first) && (actual < (best-umbral) )){
    SetBssid (apAddresses[apId]);
    updateAssociations(lastMacBssid[m_idNode], apAddresses[apId]);

    SetState (WAIT_ASSOC_RESP);
    SendAssociationRequest ();
    lastMacBssid[m_idNode] = apAddresses[apId];
    lastIdBssid[m_idNode] = apId;
}

recvApIds[m_idNode].clear();
recvUsersNumbers[m_idNode].clear();
recvDistances[m_idNode].clear();
recvApLoads[m_idNode].clear();
recvApRates[m_idNode].clear();

//Se borran los valores de los beacon
for (uint16_t i = 0; i < YansWifiPhy::beaconRSSI[m_idNode].size(); i++)
    YansWifiPhy::beaconRSSI[m_idNode][i] = -10000;

    Simulator::Schedule(MilliSeconds (m_interval), &StaWifiMac::decisionRSSI, this);
}
}
```

Código D.3: Implementación de un algoritmo de decisión simple en la clase StaWifiMac.

En los dos siguientes fragmentos de código se expone la implementación de los algoritmos de selección donde se emplean dos criterios (RSSI y carga del canal) y que se han propuesto en el trabajo como una mejora de los algoritmos sencillos que solo emplean una métrica.

```
for (uint32_t i = 0; i < recvApIds[m_idNode].size(); ++i){
    rssiAux = YansWifiPhy::beaconRSSI[m_idNode][recvApIds[m_idNode][i]];
    loadAux = recvApLoads[m_idNode][i];

    // Ponderacion de los umbrales de RSSI
    if ((rssiAux > -96) && (rssiAux <= -93+m_mixedUmbral)) //1Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 1;
    else if ((rssiAux > -93+m_mixedUmbral) && (rssiAux <= -91+m_mixedUmbral)) //2Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 2;
    else if ((rssiAux > -91+m_mixedUmbral) && (rssiAux <= -88+m_mixedUmbral)) //6Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 6;
    else if ((rssiAux > -88+m_mixedUmbral) && (rssiAux <= -87.5+m_mixedUmbral)) //9Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 9;
    else if ((rssiAux > -87.5+m_mixedUmbral) && (rssiAux <= -84.5+m_mixedUmbral)) //12Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 12;
    else if ((rssiAux > -84.5+m_mixedUmbral) && (rssiAux <= -81+m_mixedUmbral)) //18Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 18;
    else if ((rssiAux > -81+m_mixedUmbral) && (rssiAux <= -78+m_mixedUmbral)) //24Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 24;
    else if ((rssiAux > -78+m_mixedUmbral) && (rssiAux <= -73.5+m_mixedUmbral)) //36Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 36;
    else if ((rssiAux > -73.5+m_mixedUmbral) && (rssiAux <= -72+m_mixedUmbral)) //48Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 48;
    else if ((rssiAux > -72+m_mixedUmbral)) //54Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 54;
    else
        std::cout << "Algo va mal: rssi=" << rssiAux << std::endl;

    // Ponderacion de los umbrales de carga del canal
    if ((loadAux > 0.9) && (loadAux <= 1)) //1Mbps
        vectorPonderacion[i] = vectorPonderacion[i] * 1;
    else if ((loadAux > 0.8) && (loadAux <= 0.9))
        vectorPonderacion[i] = vectorPonderacion[i] * 2;
    else if ((loadAux > 0.7) && (loadAux <= 0.8))
        vectorPonderacion[i] = vectorPonderacion[i] * 3;
    else if ((loadAux > 0.6) && (loadAux <= 0.7))
        vectorPonderacion[i] = vectorPonderacion[i] * 4;
    else if ((loadAux > 0.5) && (loadAux <= 0.6))
        vectorPonderacion[i] = vectorPonderacion[i] * 5;
    else if ((loadAux > 0.4) && (loadAux <= 0.5))
        vectorPonderacion[i] = vectorPonderacion[i] * 6;
    else if ((loadAux > 0.3) && (loadAux <= 0.4))
        vectorPonderacion[i] = vectorPonderacion[i] * 7;
    else if ((loadAux >= 0.0) && (loadAux <= 0.3))
```

```
vectorPonderacion[i] = vectorPonderacion[i] * 8;

// Bonificacion al canal actual
if (recvApIds[m_idNode][i] == lastIdBssid[m_idNode]){
    vectorPonderacion[i] = vectorPonderacion[i] * m_mixedBonusUmbral;
    actual = vectorPonderacion[i];
    actualLoad = recvApLoads[m_idNode][i];
}
}

//Buscar el AP con el mejor resultado del vector
for (uint32_t i = 0; i < recvApIds[m_idNode].size(); ++i){
    if (vectorPonderacion[i] > best){
        pos = i;
        apId = recvApIds[m_idNode][i];
        best = vectorPonderacion [i];
    }
}
```

Código D.4: Implementación del algoritmo de decisión mixto1 en la clase StaWifiMac.

```
for (uint32_t i = 0; i < recvApIds[m_idNode].size(); ++i){
    rssiAux = YansWifiPhy::beaconRSSI[m_idNode][recvApIds[m_idNode][i]];
    loadAux = recvApLoads[m_idNode][i];

    vectorPonderacion[i] = m_mixedUmbral*(96+rssiAux) + m_mixedUmbral2*(1-loadAux);

    // Bonificacion al canal actual
    if (recvApIds[m_idNode][i] == lastIdBssid[m_idNode]){
        vectorPonderacion[i] = vectorPonderacion[i] * m_mixedBonusUmbral;
        actual = vectorPonderacion[i];
        actualLoad = recvApLoads[m_idNode][i];
    }
}

//Buscar el AP con el mejor resultado del vector
for (uint32_t i = 0; i < recvApIds[m_idNode].size(); ++i){
    if (vectorPonderacion[i] > best){
        pos = i;
        apId = recvApIds[m_idNode][i];
        best = vectorPonderacion [i];
    }
}
```

Código D.5: Implementación del algoritmo de decisión mixto2 en la clase StaWifiMac.

El resumen de todas las modificaciones realizadas en el simulador ns-3 se podría resumir en el siguiente diagrama de estados de la figura D.1 donde se puede observar la combinación tanto el soporte para múltiples canales como el sistema de asociación implementado y explicado en este anexo.

Si comparamos el gráfico siguiente con la figura B.2 que representaba el diagrama de estados original donde no se soportaba ni múltiples canales ni reasociaciones entre APs se pueden observar las enormes diferencias tanto a nivel de funcionalidad como en cuanto a complejidad entre un modelo y otro.

Por último, y debido a que se pueden personalizar completamente el formato de los Beacon a partir en la clase `MgtBeaconHeader`, la situación ideal de simulación sería incluir las métricas de números de usuarios asociados, carga del canal y tasa cursada en el mismo paquete Beacon tal y como se especifica en el estándar 802.11k (expuesto en el anexo F) para luego extraer estas estadísticas en la clase `StaWifiMac`. De esta forma, estaríamos simulando completamente lo que ocurriría en caso de que implementáramos la solución propuesta en un sistema real. Sin embargo, el impacto que habría a nivel de rendimiento comparado con la alternativa empleada en el simulador sería mínimo ya que la única diferencia sería que el paquete de los Beacon aumentaría ligeramente al incluir estos datos.

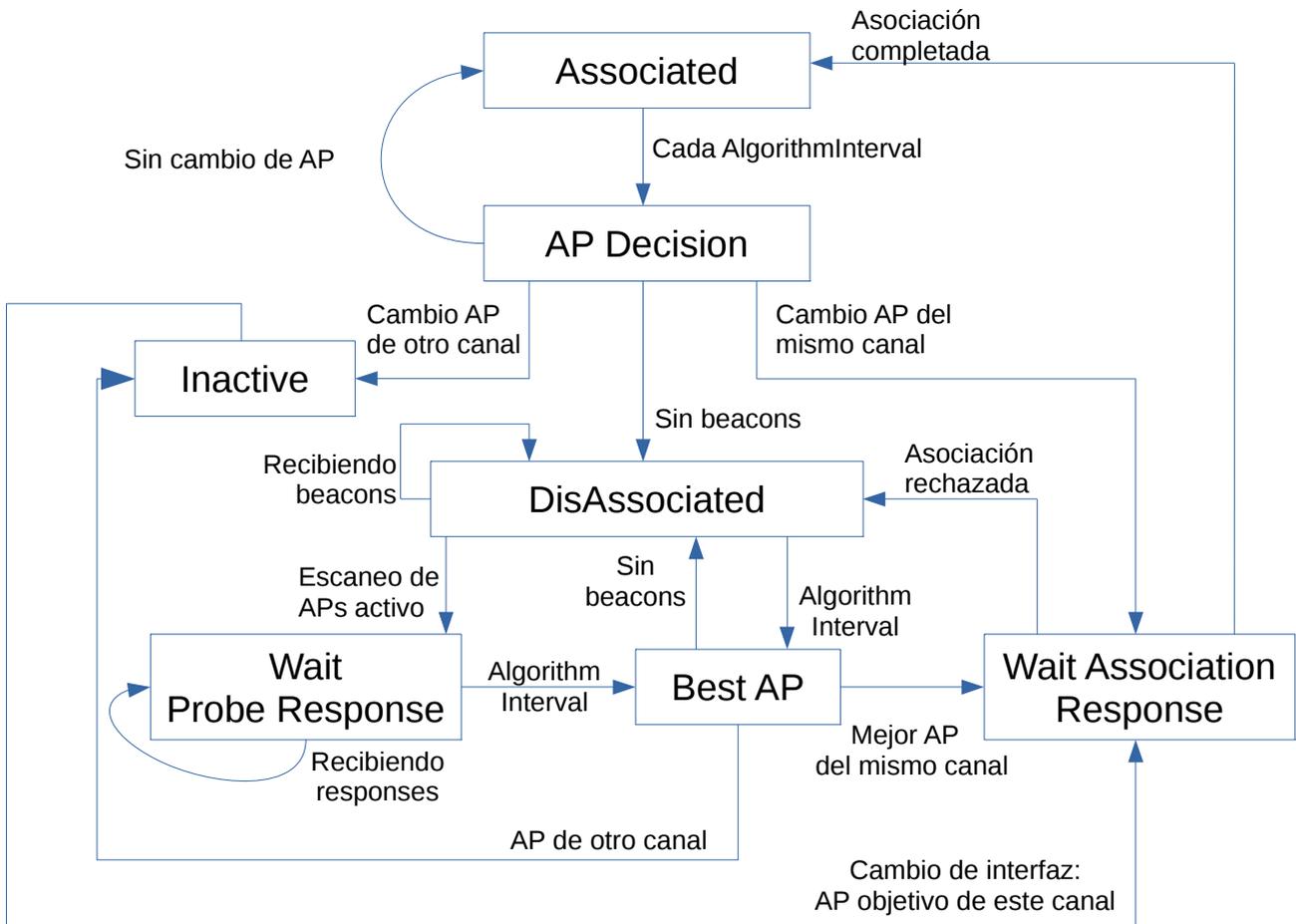


Figura D.1: Máquina de estados propuesta para las estaciones móviles en modo infraestructura.



## Anexo E. Estándares 802.11

Desde que en 1997 fuera lanzada de forma oficial la primera versión de la familia 802.11 el grupo de trabajo del IEEE dedicado a las redes WLAN ha lanzado más de 25 normas que actualizan o corrigen los anteriores estándares. De entre todas ellas, hay 5 versiones que destacan sobre el resto al incluir grandes revisiones y avances al introducir nuevos esquemas de modulación y codificación (MCS) y otras técnicas que permiten aumentar las capacidades de transmisión.

### E.1 Normas 802.11 de nivel físico

Las principales novedades entre estos estándares se encuentran en su capa física ya que las mayores diferencias se encuentran en las frecuencias, las modulaciones y los anchos de bandas que soportan. En la tabla E.1 se puede ver un resumen de las diferencias entre las normas más importantes que ya han sido lanzadas oficialmente por el IEEE.

	Norma				
	802.11a	802.11b	802.11g	802.11n	802.11ac
Año	1999	1999	2003	2009	2013
Banda [GHz]	5	2,4	2,4 / 5	2,4 / 5	5
Tasa máxima [Mbps]	54	11	54	150 (600)	867 (6930)
Modulaciones	OFDM	CCK, DSSS	CCK, DSSS, OFDM	OFDM	OFDM
Soporte MIMO	No	No	No	Si (4)	Si (8)
Ancho de banda [MHz]	20	22	20 / 22	20 / 40	20 - 160

Tabla E.1: Especificaciones de nivel físico de estándares 802.11

Las principales características y novedades introducidas por los anteriores estándares son las siguientes:

- IEEE 802.11a-1999  
Fue la primera revisión junto con el 802.11b del estándar 802.11 y se caracterizaba por emplear la banda de 5GHz y modulación OFDM. Su gran ventaja es que presenta hasta 13 canales sin solape aunque no fue suficiente para que fuera empleado masivamente, siendo su uso anecdótico en la actualidad.

- IEEE 802.11b-1999  
Fue lanzado junto con 802.11a, convirtiéndose rápidamente en el estándar preferido por los fabricantes de dispositivos Wi-Fi. Empleaba la modulación DSSS definida del primer estándar pero aumentando la máxima tasa física de los 2Mbps hasta los 11Mbps.
- IEEE 802.11g-2003  
Diseñado desde sus orígenes para ser completamente retrocompatible con 802.11b definiendo para ello cuatro posibles capas físicas, algunas conocidas como *Extended Rate Physical* (ERP), que combinan las nuevas modulaciones OFDM (heredadas del 802.11a) y permiten aumentar la tasa física hasta los 54Mbps con las ya soportadas en el 802.11b.
- IEEE 802.11n-2009  
Norma que volvió a ser diseñada para ser retrocompatible con las recomendaciones anteriores pero esta vez también con la 802.11a ya que también contempla el uso de la banda de 5GHz. Las principales novedades son la introducción de antenas MIMO, el aumento opcional del ancho de banda y algunas medidas encaminadas a aumentar la eficiencia de transmisión.
- IEEE 802.11ac-2013  
Elimina la opción de trabajar en la banda de 2,4GHz y añade nuevas esquemas de modulación de alta densidad (256Q-AM). Además, continúa con la tendencia marcada por el 802.11n aumentando el número de *streams* espaciales y aumentando los posibles anchos de banda pero, al contrario que la anterior norma, su penetración en el mercado está siendo muchísimo más lenta.

## **E.2 Otras normas 802.11 importantes**

Algunas de las normas IEEE 802.11 destinadas a aumentar las capacidades de las redes WLAN relacionadas con la calidad de servicio y la optimización de itinerancia de las estaciones son las siguientes:

- IEEE 802.11e-2005  
Norma destinada a aportar QoS para optimizar los servicios en tiempo real. Introduce el concepto de *Hybrid Coordinator* (HC), que es un punto de acceso que controla el acceso al medio de las STAs. Extiende los dos métodos de acceso al medio existentes:
  - **EDCA: *Enhanced distributed channel access***  
Mejora al antiguo sistema DCF incorporando un nuevo intervalo denominado AIFS (*Arbitration Inter-frame Space*) que prioriza entre los 4 tipos de tráfico que define. Además añade un periodo de contienda donde el acceso al medio es libre y las STAs pueden intentar transmitir todo que puedan.
  - **HCCA: *HCF Controlled Channel Access***

Mejora a PCF, siendo igualmente opcional en los dispositivos 802.11e. Aporta diferenciación entre clases y flujos de tráfico, permitiendo diferenciar entre sesiones dentro de las colas de una misma STA.

Introduce otras nuevas funcionalidades como la posibilidad de eliminar los ACK del nivel MAC y la opción *Direct Link Setup* (DLS) que permite el intercambio de tramas directamente entre STA sin necesidad de pasar por el AP.

- IEEE 802.11f-2003  
Norma que define el *Inter-Access Point Protocol* (IAPP), protocolo destinado a la comunicación entre los AP, sin importar su fabricante, para la gestión del roaming y el balanceo de la carga.
- IEEE 802.11k-2008  
Esta recomendación incorpora nuevos mecanismos para optimizar la gestión de los recursos radio. Esta versión se encuentra explicada con mayor detalle en el anexo F.
- IEEE 802.11r-2008  
Norma conocida también como *Fast BSS Transition* que introduce mejoras en la velocidad y la seguridad en los traspasos de las STAs aportando un nuevo concepto de roaming donde las primeras comunicaciones con el futuro AP se realizan antes de abandonar la antigua asociación.
- IEEE 802.11ae-2012  
Incorpora nuevas mejoras a la gestión de QoS, expandiendo el antiguo 802.11e. Los nuevos mecanismos están orientados a priorizar las tramas de gestión, desarrollando para ello una nueva política de priorización.

### **E.3 Modelo de gestión 802.11**

La estructura del modelo que establece los niveles de gestión está muy relacionado con la estructura de capas que 802.11 presenta en los niveles inferiores físico y MAC [36]. En el caso del nivel físico se encuentran definidas dos subcapas.

- PMD (Physical Medium Dependent) es la subcapa más inferior del modelo y la encargada de transmitir el flujo de bits físicamente que recibe de la subcapa superior PLCP a través de la antena o antenas de las que disponga el dispositivo o de recibirlo adecuadamente para pasarlo posteriormente la subcapa PCLP. Para realizar estas dos tareas de forma correcta debe usar la modulación correspondiente según se define en el estándar y que varía según la especificación a/b/g/n que se esté empleando y de lo que le esté indicando la subcapa PLCP.
- PLCP (*Physical Layer Convergence Procedure*): es la subcapa que une una capa física PMD con el nivel MAC. Sus funciones consisten en añadir una cabecera propia e incluir un preámbulo que facilite la sincronización de las transmisiones. La cabecera incorpora campos

importantes e imprescindibles para la recepción de las tramas como la longitud en microsegundos y la tasa empleada. Además también es la subcapa que añade o sustrae el código convolución para la detección de errores.

Las especificaciones del nivel físico del estándar 802.11 incluyen diferentes entidades de gestión que se concretan y definen en dos subcapas de gestión MLME para el nivel de enlace y PLME para el nivel físico. Además de sus propias funcionalidades, estas subcapas también proporcionan interfaces de gestión a través de los cuales sus funciones de gestión pueden ser invocadas desde otras subcapas o desde aplicaciones definidas dentro del SME (*Station Management Entity*).

- PLME (*PHY Layer Management Entity*)
  - Es la capa de gestión que tiene acceso a los parámetros radio disponibles en el hardware del dispositivo.
  - Los parámetros que se encuentran almacenados en la MIB de la PLME son los de nivel físico, es decir, el canal actual, la potencia disponible de transmisión, el modo CCA, número de símbolos en una trama...
  
- MLME (*MAC Layer Management Entity*)
  - Es la capa de gestión del nivel MAC y se encarga de la gestión de los interfaces con el SME y el acceso a la información de la información del nivel físico a través del PLME SAP. Además también tiene la responsabilidad de gestionar su propia base datos MIB que contiene los objetos y las variables de la capa MAC.
  - Los mecanismos que se definen en esta subcapa están relacionados con la gestión de la potencia, funciones de escaneo, sincronización con los AP (asociación, reasociación y autenticación), solicitud de medidas...
  - Tanto en este caso, como en el del PLME, el estándar define únicamente los servicios que los interfaces SAP (*Service Access Point*) deben soportar, pero nunca su implementación específica.
  
- SME (*Station Management Entity*)
  - El SME es una capa independiente que podría definirse como un plano de gestión que funciona de forma separada al resto y cuyas funciones no están especificadas en el estándar. Sin embargo, las funciones más comunes son la reunión de la información necesaria del resto de subcapas de gestión así como la configuración de estas.
  - En esta capa es donde se pueden implementar los distintos protocolos de gestión que determinarían el comportamiento tanto de las STAs como de las APs ante las asociaciones, reasociaciones ...
  - Aunque su definición no se establezca en el estándar, es necesario que todos los nodos tengan una instancia SME con unas funciones mínimas que permitan garantizar el correcto funcionamiento del nivel MAC.

Las tres entidades de gestión anteriores pueden interactuar de diferentes maneras como se puede observar en la figura E.1 donde se representa la relación entre las distintas entidades de gestión. Como ya se ha comentado, el estándar solo define explícitamente algunas de las posibles

interaccionares y en este caso son las que se realizan a través del SAP donde las primitivas a intercambiar ya están definidas. En el caso de las relaciones entre las capas MAC-MLME y las capas PLCP/PMD-PLME no hay ninguna directiva en el estándar que establezca como deben ser las comunicaciones entre ellas.

En el caso de la PLME SAP las primitivas que soportan son exactamente las mismas y se podría considerar como un único interfaz que puede ser usado directamente desde la MLME o la SME por lo que la capa de gestión del nivel físico puede ser controlada tanto desde el nivel MAC como desde la SME.

Toda la información de gestión especificada en el estándar debe encontrarse almacenada en una MIB de forma independiente para cada capa, pudiendo el gestor de la red ampliarla con más información. Para acceder a los diferentes atributos de las MIB, la comunicación se realiza a partir de unas primitivas genéricas ya definidas y que pueden ser GET o SET en función de si queremos consultar o modificar el atributo. Estas dos primitivas son conocidas como Request y llevan asociadas siempre una respuesta definida como Confirm.

Subcapas que contienen extensiones de RRM

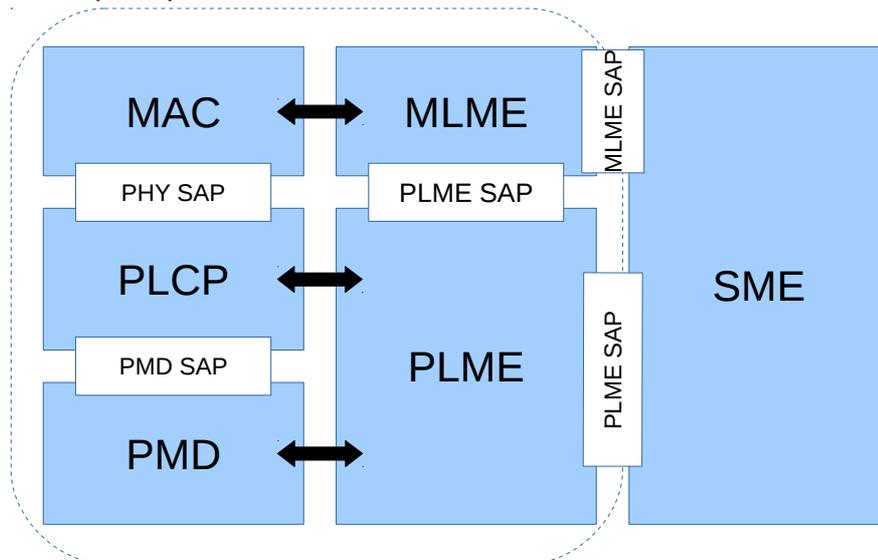


Figura E.1: Modelo de gestión 802.11



## Anexo F. Estándar IEEE 802.11k.

IEEE 802.11k-2008 [3] es una recomendación de la familia IEEE 802.11 que extiende los mecanismos para la gestión de los recursos radio (RRM por sus siglas en inglés, *Radio Resource Management*) en redes de área local (WLAN). El objetivo principal de esta recomendación es la optimización en la gestión de las redes inalámbricas para incrementar el rendimiento global.

La gestión de recursos radio es una de las claves que permitirá el correcto funcionamiento de la siguiente generación de redes inalámbricas WLAN. La RRM aborda algunos de los problemas existentes que surgen al usar entornos con bandas radio no licenciadas como las ISM y aporta soluciones para lograr los elevados requisitos de las tecnologías emergentes. Además, la RRM provee otras medidas que aportan conocimiento sobre el entorno radio que permiten aumentar la fiabilidad de las redes más allá de la mejora de la potencia y de la tasa global.

La información y las medidas existentes sobre las WLAN antes de la definición del estándar 802.11 se antojaban inadecuadas para avanzar a las redes de nueva generación. El enfoque propuesto en la medida de los recursos radio está dirigido a añadir medidas que amplíen la capacidad, la fiabilidad y las posibilidades de mantener adecuadamente la redes WLAN aportando estas medidas a las capas superiores de la pila de protocolos.

### **F.1 Nuevos formatos de las tramas de gestión**

Para soportar los nuevos mecanismos de RRM los formatos de las tramas de gestión Beacon, asociación, reasociación y Probe tanto en su forma request como en response han sido modificados en el estándar 802.11k para contener diferentes campos, todos ellos opcionales, que contengan las diferentes métricas que permitan tanto a los puntos de acceso como a los nodos móviles actuar de forma inteligente en la gestión de los recursos radio.

La disponibilidad de los diferentes campos varía en función de la trama de gestión que se esté transmitiendo. En el caso de las tramas Beacon o Probe Response los nuevos campos que se han introducido son los siguientes:

- Informe de canales (*AP Channel Report*)
  - El informe está compuesto por una lista de canales donde la STA ha escuchado y ha encontrado AP donde puede asociarse.
  - La lista de canales contiene un número variable de octetos donde cada octeto define de forma independiente cada número de canal.

- Retardo de acceso medio en el BSS (BSS Average Access Delay)

Este elemento contiene el retardo de acceso medio en el AP, que es una medida de la carga en el BSS. Esta medida está disponible tanto en los puntos de acceso con QoS activada y sin ella.

  - El campo es un número entero de 8 bits que representa el nivel relativo de carga en el AP. Un AP con un valor bajo indica que hay más capacidad disponible que en un AP con un nivel alto. El único número que no respeta esta norma es el valor reservado 255 que indica que el AP no está transmitiendo ninguna trama DCF o EDCA.
  - Así, los valores útiles son de 0 a 252 y son una escala ya definida y no lineal que representa el tiempo desde que la trama DCF o EDCA (si el AP tiene habilitada la QoS) está lista para transmitir hasta que comienza la transmisión. El valor mínimo es 0 que indica que el retardo es menor a 8 microsegundos y el máximo retardo medible es de 24,576 milisegundos.
  
- Información de antena (Antenna Information)
  - El campo de Id de antena contiene un número que identifica las antenas relevantes según el tipo de trama donde se incluya. Cuando este campo es incluido dentro de un Beacon u un Probe Response identifica la antena o antenas que se han usado para transmitir la trama. En caso de que este campo se incluya en un informe de medidas, este número identifica la antena o antenas que se han empleado para tomar las medidas.
  
- Capacidad de admisión disponible en el BSS (BSS Available Admission Capacity)
  - Este elemento solo se encuentra disponible en los AP con soporte para QoS y contiene una lista de campos con la capacidad de admisión disponible para las diferentes prioridades de usuario y categorías de acceso definidas en el estándar 802.11e.
  - Este elemento es útil para los roaming donde una STA asociada a un AP sin QoS va a conectarse a otro con soporte para QoS y que por lo tanto es muy probable que acepte sus futuras tramas de tráfico. Sin embargo, sólo es una medida orientativa ya que no representa ninguna garantía de que el HC admita y tramite las demandas de tráfico que la STA necesita.
  
- Retardo de acceso de las AC (Access Category) en el BSS (BSS AC Access Delay)
  - Este elemento contiene un campo que presenta un formato con cuatro subcampos, donde cada uno representa el retardo medio de acceso que tienen cada una de las cuatro colas EDCA o categorías de acceso (AC) definidas en el estándar 802.11e.
  - Cada uno de los campos es un número escalar de 8 bits cuya representación y escala es exactamente igual que la definida en el campo BSS Average Access Delay.
  
- Información sobre la transmisión de la piloto de medición (Measurement Pilot Transmission Information)
  - Este elemento contiene un campo que contiene el intervalo que indica el número de unidades temporales, medidas en microsegundos, entre los que el AP generará una trama piloto.

- Múltiple BSSID (Multiple BSSID)
  - Este campo está relacionado con el concepto “conjunto de múltiples BSSID” que también se define en este estándar y que se caracteriza por tener las siguientes características:
    - Todos sus elementos deben usar la misma antena, canal y “*regulatory class*”.
    - El tamaño del grupo está definido por  $2^n$ , donde 'n' puede tener hasta un valor de 46.
    - Todos los BSS que formen parte deben tener configurado los mismos 48 MSB en sus BSSID.
  - Lo que se transmite en el campo múltiple BSSID es el valor de la 'n', por lo que no se transmite explícitamente el número exacto de BSS que pertenecen al grupo.
  
- Capacidades de RRM activadas (RRM Enabled Capabilities)
  - Este elemento indica las características de mediciones radio que soporta el dispositivo.
  - El formato de representación son 5 octetos (solo se usan los 33 más significativos, el resto están reservados) donde según el valor de cada bit se indica si se soporta o no el mecanismo de medición o de informe.
  - Algunas de las características más empleadas son las siguientes:
    - Link Measurement capability
    - Neighbor Report capability
    - Parallel Measurements capability
    - Repeated Measurements capability
    - Beacon Passive Measurement capability
    - Beacon Active Measurement capability
    - Beacon Table Measurement capability

En caso de que la trama sea una respuesta de un AP a la petición de asociación de un STA, los campos que se incluirán son, a parte del RRM Enabled Capabilities, los dos siguientes:

- RCPI (Received Channel Power Indicator)
  - Elemento que contiene la potencia con la que se ha recibido la trama en la STA.
  
- RSNI (Received Signal to Noise Indicator)
  - Elemento que contiene la señal a ruido que ha calculado la STA a partir de la potencia con la que ha recibido la trama y el ruido más interferencia que ha detectado.
  - El indicador, de 8 bits, tiene como valores extremos -10dB y 117dB por lo que tiene una resolución de 0,5 dB.

## **F.2 Mecanismos y medidas de RRM**

El estándar 802.11k define varios medios [37][38], algunos ya existentes y otros nuevos, a partir de los cuales obtener las distintas medidas radio. Algunos de estos medios permiten a la STA conseguir información localmente a través de sus propios interfaces de gestión mientras que otros

contemplan que las STA puedan solicitar una o varias medidas explícitas a otras STAs. Todo el conjunto de RRM está destinado a poder ofrecer información a aplicaciones y capas de nivel.

Para lograr los exigentes requisitos de movilidad que presentan las tecnologías como VoIP o streaming de vídeo se pueden emplear algunas medidas radio como la carga del canal o el informe de vecinos, que permitirían recolectar información antes de realizar el handover, lo que puede aumentar la velocidad del roaming entre celdas con el mismo ESSID. Accediendo y usando esta información, tanto las STAs como los APs podrán tomar decisiones inteligentes para buscar la forma más efectiva de emplear el espectro, la potencia y el ancho de banda disponible para cubrir sus necesidades.

Los siguientes mecanismos de medida proporcionan a las STAs y APs la capacidad de gestionar y consultar su entorno radio para realizar evaluaciones adecuadas sobre su estado y la eficiencia con la que se está usando. Este debe ser el primer paso para la creación de redes WLAN inteligentes capaces de tomar las decisiones que garanticen transiciones rápidas entre APs, la conectividad en redes malladas y la gestión del entorno radio para todos los dispositivos inalámbricos.

- Beacon  
El par de mensajes request/report de Beacon permite a una STA solicitar a otra STA una lista de los puntos de acceso que él puede recibir en un uno o varios canales. Esta medida se puede realizar de tres formas: modo activo o pasivo (con un funcionamiento similar al escaneo mediante el envío de tramas Probe) donde durante un tiempo determinado la STA monitoriza el canal especificado y guarda los Beacon, los Probe response y las tramas piloto con su respectivo RCPI. La última opción es acceder directamente a una tabla con los Beacons recibidos sin ninguna información adicional de RRM más allá del SSID/BSSID.
- Measurement Pilot  
Es un mecanismo donde solo se generan mensajes de tipo report y consiste en una trama de medida piloto que es de un tamaño reducido inferior a la de un Beacon y que es transmitida periódicamente por un AP en un intervalo más pequeño que el definido para los Beacon. Sus funciones son parecidas y su principal propósito es ayudar a las STA cuando escaneen el canal.
- Frame  
El par de mensajes request/report “frame” devuelve una imagen con todo el tráfico que la STA ha recibido contando el número de tramas recibidas. El informe distingue entre las diferentes fuentes de tráfico y contiene el número de tramas recibidas, el RCPI medio y el BSSID del equipo transmisor.
- Carga del canal  
El par de mensajes request/report de carga del canal devuelve una medida con la utilización del canal observada por la STA.
- Histograma del ruido  
Los mensajes request/report de histograma del ruido tienen como resultado un histograma con la potencia medida de fuentes de ruido externas. Esto se consigue analizando el canal

cuando el protocolo de acceso al medio determina que el canal está en reposo y la STA no está ni transmitiendo ni recibiendo ninguna trama.

- Estadísticas del STA

El par request/report de estadísticas del STA genera un conjunto de valores agrupados en dos categorías: contadores de STA y retardo de acceso medio de BSS.

- Los contadores de STA incluye el número de fragmentos transmitidos unicast y multicast, número de tramas fallidas y reintentos, número de tramas duplicadas, RTS exitosos y fallidos, ACKs fallidos, fragmentos unicast y multicast recibidos correctamente, errores de FCS y número total de tramas transmitidas.
- El grupo de retardo medio en el acceso contiene el retardo medio de acceso del AP y para cada una de las diferentes categorías de acceso, un contador con el número de STAs asociadas y la utilización del canal.

- Localización

El par de mensajes request/report de localización devuelven la posición de una STA en función de su latitud, longitud y altitud, diferenciando en esta última varios tipos de alturas y resoluciones, por ejemplo, en función del número de pisos. Los nodos móviles pueden solicitar tanto su propia posición como la de otra STA.

- Pausa en las medidas

Consiste en un mensaje request, que en esta ocasión no genera un report, que permite introducir un cierto retraso en la ejecución de las medidas individuales cuando se van a realizar varias medidas sobre una misma trama.

- Informe de vecinos

En este caso el request sobre los vecinos es enviado a un AP, que devuelve un informe sobre los AP que tiene a su alrededor y que serían candidatos para realizar un handover. El informe enviado por el AP contiene la información definida en una tabla ya estandarizada y almacenada en una MIB.

- Medidas del enlace

Los mensajes request/report de medidas del enlace ofrecen una medida que indica la calidad instantánea de un enlace, intercambiando las características del canal RF medidas en un enlace existente entre una STA y otro STA.

- Medidas de la categoría/flujo transmitido

Las medidas de la categoría o flujo transmitido son un par de mensajes request/report que permiten a un STA con calidad de servicio solicitar a otra STA con calidad de servicio el estado en el que se encuentra un flujo de tráfico en curso, proporcionando las métricas para medir el flujo de tráfico desde el lado del transmisor. Se pueden configurar unos umbrales para que se generen informes de forma automática.

En la figura E.2 está representado el diagrama de secuencia donde se representa el intercambio de mensajes que existe tanto entre dos estaciones como entre las diferentes capas del módulo de gestión de un mismo dispositivo para el acceso a las MIB donde consultar las medidas solicitadas.

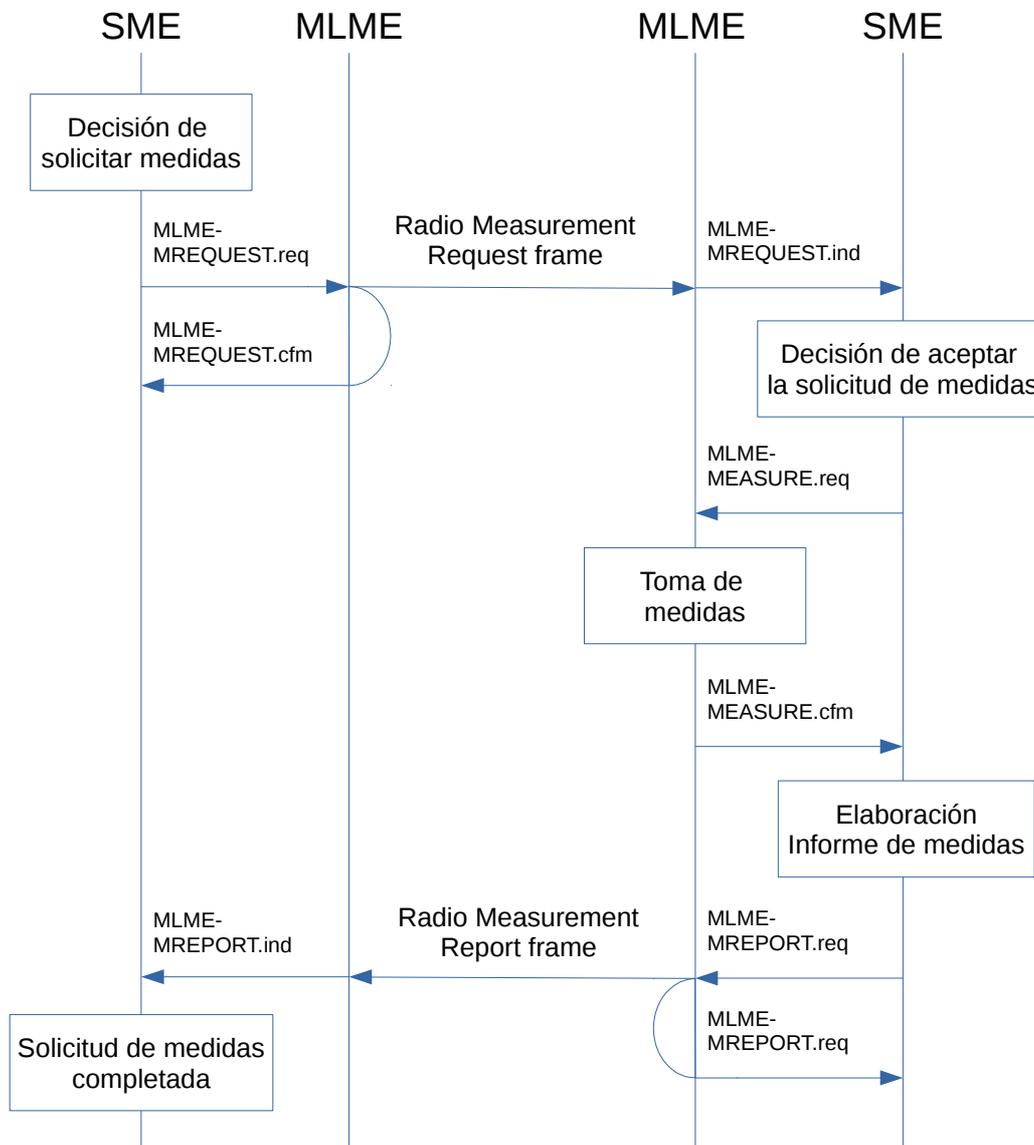


Figura E.2: Diagrama de secuencias de una solicitud de medidas



## Anexo G. Cronología del trabajo.

Los pasos o actividades que se han llevado a cabo hasta la finalización de este trabajo con las fechas estimadas de comienzo, finalización y su duración se pueden ver tanto en la tabla G.1 como en la figura G.1.

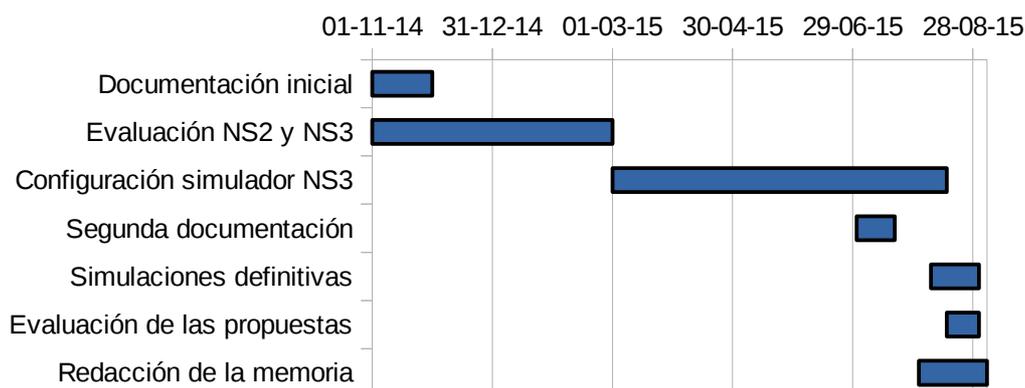


Figura G.1: Diagrama de Gantt

Actividad	Fecha inicio	Fecha final	Duración
Documentación inicial sobre adaptación de potencia y tasa en 802.11	Noviembre 2014	Diciembre 2015	30 horas
Evaluación ns-2 y ns-3	Noviembre 2014	Marzo 2015	50 horas
Aprendizaje, configuración y modificación del simulador NS3	Marzo 2015	Agosto 2015	300 horas
Segunda documentación	1 Julio 2015	20 Julio 2015	25 horas
Simulaciones definitivas	7 Agosto 2015	31 Agosto 2015	*
Evaluación de las propuestas	24 Agosto 2015	31 Agosto 2015	25 horas
Redacción de la memoria	1 Agosto 2015	4 Septiembre 2015	100 horas

Tabla G.1: Cronología del trabajo

(\*) La duración de las simulaciones definitivas no se ha contabilizado ya que gracias a que el escenario se configuró para poder emplear scripts que configuraran dinámicamente el escenario y el comportamiento de la simulación no ha sido necesario recoger los resultados uno por uno. Sin embargo, si hubiera que decir el número de horas que ha habido simulaciones funcionando, serían más de 500 horas y si se tiene en cuenta que ha habido varios ordenadores (hasta 3 a la vez más las máquinas virtuales de la nube) el tiempo total de simulación superaría fácilmente las 1800 horas.



# Bibliografía

- [1] ISO/IEC 8802-11:2005/AMD4 [IEEE Std 802.11g-2003] Information technology-- Local and metropolitan area networks-- Part 11: Wireless LAN Medium Access Control (Mac) and Physical Layer (PHY) Specifications--Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band," in *ISO/IEC 8802-11:2005/Amd.4:2006(E) IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11-1999)*, vol., no., pp.1-83, Aug. 15 2006
- [2] IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," in *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, vol., no., pp.1-565, Oct. 29 2009
- [3] IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs," *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, vol., no., pp.1,244, June 12 2008
- [4] Yihong Zhou; Nettles, S.M., "Balancing the hidden and exposed node problems with power control in CSMA/CA-based wireless networks," in *Wireless Communications and Networking Conference, 2005 IEEE* , vol.2, no., pp.683-688 Vol. 2, 13-17 March 2005
- [5] An Chan; Soung Chang Liew, "Merit of PHY-MAC Cross-Layer Carrier Sensing: A MAC-Address-based Physical Carrier Sensing Scheme for Solving Hidden-Node and Exposed-Node Problems in Large-Scale Wi-Fi Networks," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on* , vol., no., pp.871-878, 14-16 Nov. 2006
- [6] Kamerman, Ad; Monteban, Leo, "WaveLAN®-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal* , vol.2, no.3, pp.118,133, Summer 1997
- [7] M. Lacage, M.H. Manshaei, and T. Turletti. "IEEE 802.11 rate adaptation: a practical approach". Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pages 126–134, 2004
- [8] Maguolo, F.; Lacage, M.; Turletti, T., "Efficient collision detection for auto rate fallback algorithm," *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, vol., no., pp.25,30, 6-9 July 2008
- [9] S.Kim, S.Choi and D.Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs" in Proc. IEEE INFOCOM, Barcelona, Spain, April 2006.

- [10] G. Holland, N. Vaidya, and P. Bahl. "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks". In Proceeding ACM MOBICOM, July 2001.
- [11] Hyun-Woo Lee; Se-han Kim; Won Ryu; Chong-ho Yoon, "Performance of an efficient method for association admission control in public wireless LAN systems," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th* , vol.7, no., pp.5049-5053 Vol. 7, 26-29 Sept. 2004
- [12] Lei Du; Yong Bai; Lan Chen, "Access Point Selection Strategy for Large-Scale Wireless Local Area Networks," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE* , vol., no., pp.2161-2166, 11-15 March 2007
- [13] Nam Nguyen; Sato, T., "A Proposal for Dynamic WLAN Selection for Mobile Data Offloading in Heterogeneous Network," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th* , vol., no., pp.1-5, 18-21 May 2014
- [14] Jyh-Cheng Chen; Tuan-Che Chen; Tao Zhang; van den Berg, E., "WLC19-4: Effective AP Selection and Load Balancing in IEEE 802.11 Wireless LANs," in Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE , vol., no., pp.1-6, Nov. 27 2006-Dec. 1 2006
- [15] T. Korakis, O. Ercetin, S. Krishnamurthy, L. Tassiulas, and S. Tripathi, "Link Quality based Association Mechanism in IEEE 802.11h Compliant Wireless LANs," in Proc. RAWNET'06, Boston, MA, Apr. 2006, pp. 725–730.
- [16] Yapeng Wang; Cuthbert, L., "A Global Optimisation Algorithm for 802.11 WLAN Radio Resource Management," in *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on* , vol., no., pp.2144-2147, 21-25 Sept. 2007
- [17] O. Brickley, S. Rea, and D. Pesch, "Load balancing for QoS enhancement in IEEE802.11e WLANs using cell breathing techniques," in IFIP Mobile and Wireless Communication Networks Conf., Marrakech, Morocco, Sept. 2005.
- [18] Garcia, E.; Vidal, R.; Paradells, J., "Cooperative load balancing in IEEE 802.11 networks with cell breathing," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on* , vol., no., pp.1133-1140, 6-9 July 2008
- [19] D. Faria, Modeling Signal Attenuation in IEEE 802.11 Wireless LANs, Computing Science Department Stanford University.
- [20] Cebula, S. L., Ahmad, A., Graham, J. M., Hinds, C., Wahsheh, L. A., Williams, A. T., and DeLoatch, S. J., "Empirical Channel Model for 2.4GHz IEEE 802.11 WLAN." In Proceedings of the 2011 International Conference on Wireless Networks, July 2011.
- [21] Akl, R., Tummala, D., and Li, X. "Indoor Propagation Modeling at 2.4 GHz for IEEE 802.11 Networks". In Proceedings of the 6 th IASTED International Multi-Conference on Wireless and Optical Communications, Banf, AB, Canada, 2006.

- 
- [22] Cisco Aironet 1240AG Series 802.11A/B/G Access Point Data Sheet: [http://www.cisco.com/c/en/us/products/collateral/collaboration-endpoints/unified-ip-phone-7900-series/product\\_data\\_sheet0900aecd8031c844.html](http://www.cisco.com/c/en/us/products/collateral/collaboration-endpoints/unified-ip-phone-7900-series/product_data_sheet0900aecd8031c844.html)
- [23] BOE-A-2010-2719.pdf. <http://boe.es/boe/dias/2010/02/19/pdfs/BOE-A-2010-2719.pdf>
- [24] M. Khosroshahy. *Study and Implementation of IEEE 802.11 Physical Channel Model in YANS (NS3 prototype) Network Simulator*, 2006.
- [25] Pei and Henderson. Validation of OFDM error rate model in ns-3. Boeing Research and Technology, May 2009.
- [26] Pei and Henderson. Validation of ns-3 802.11b PHY model. Boeing Research and Technology, May 2009.
- [27] G.711: Pulse code modulation (PCM) of voice frequencies <http://www.itu.int/rec/T-REC-G.711>
- [28] ns-3 The Discrete-event network simulator. <https://www.nsnam.org/>
- [29] ns-2 - The Network Simulator <http://www.isi.edu/nsnam/ns/>
- [30] ns-3 Tutorial. <https://www.nsnam.org/docs/tutorial/html/>
- [31] ns-3 Documentation. <https://www.nsnam.org/doxygen/index.html>
- [32] Design Documentation. Wi-Fi Module. <https://www.nsnam.org/docs/models/html/wifi-design.html>
- [33] Yet Another Network Simulator. <http://cutebugs.net/files/wns2-yans.pdf>
- [34] <http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel/>
- [35] Ernst, J.B.; Kremer, S.C.; Rodrigues, J.J.P.C., "A Wi-Fi Simulation Model Which Supports Channel Scanning across Multiple Non-overlapping Channels in NS3," *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on* , vol., no., pp.268,275, 13-16 May 2014
- [36] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)* , vol., no., pp.1,1076, June 12 2007
- [37] E. García Villegas, Self-Optimization of Radio Resources on IEEE 802.11 Networks, July 2009.

[38] 802.11r, 802.11k, and 802.11w Deployment Guide, Cisco IOS-XE Release 3.3 – Chapter 3  
[http://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/5700/software/release/ios\\_xe\\_3/3/11rkw\\_DeploymentGuide/b\\_802point11rkw\\_deployment\\_guide\\_cisco\\_ios\\_xe\\_release33.pdf](http://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/5700/software/release/ios_xe_3/3/11rkw_DeploymentGuide/b_802point11rkw_deployment_guide_cisco_ios_xe_release33.pdf)

