



ESTUDIOS DE INGENIERÍA
DE TELECOMUNICACIÓN

PROYECTO DE FIN DE CARRERA

**GESTIÓN DE RECURSOS PARA
SERVICIOS DE TIEMPO REAL SOBRE
REDES WLAN**

CURSO: 2008/09

José Antonio Rodríguez Salas

Julio de 2009



ESTUDIOS DE INGENIERÍA DE TELECOMUNICACIÓN

GESTIÓN DE RECURSOS PARA SERVICIOS DE TIEMPO REAL SOBRE REDES WLAN

REALIZADO POR:

José Antonio Rodríguez Salas

DIRIGIDO POR:

D. Jorge Navarro Ortiz

DEPARTAMENTO:

TEORÍA DE LA SEÑAL, TELEMÁTICA Y COMUNICACIONES
Universidad de Granada

Granada, Julio de 2009

ACTA DE VALORACIÓN DEL PROYECTO

El tribunal constituido para la evaluación del proyecto PFC titulado:

Gestión de recursos para servicios de tiempo real sobre redes WLAN

Realizado por el alumno: ***José Antonio Rodríguez Salas***

Y dirigido por el tutor: ***D. Jorge Navarro Ortiz***

Ha resuelto asignarle la calificación de:

☐

SOBRESALIENTE (9-10 puntos)

☐

NOTABLE (7-8.9 puntos)

☐

APROBADO (5-6.9 puntos)

☐

SUSPENSO

Con la nota de:

puntos.

NOMBRE DE LOS MIEMBROS DEL TRIBUNAL:

Presidente: D.

Vocal: D.

Secretaria: D^a.

Presidente suplente:

Vocal suplente:

El Presidente: _____

La Secretaria: _____

El Vocal: _____

Fecha de la lectura y defensa: Granada, a _____ de _____ de 20____

GESTIÓN DE RECURSOS PARA SERVICIOS DE TIEMPO REAL SOBRE REDES WLAN

(José Antonio Rodríguez Salas)

PALABRAS CLAVE

Acceso Coordinado y Distribuido Mejorado (EDCA), Mecanismo de Acceso al Medio Controlado por HCF (HCCA), Coordinador Híbrido (HC), *Network Simulator 2 (NS2)*, Calidad de Servicio (QoS), Planificador de Paquetes (*Scheduler*), Redes de Área Local Inalámbrica (WLAN), *Wireless Timed Token Protocol (WTTT)*.

RESUMEN

Este proyecto se centra en el análisis y mejora de las redes WLAN para soportar la calidad de servicio requerida por los servicios de entretenimiento en el hogar, como la telefonía y el vídeo bajo demanda. Con este fin se ha utilizado un estándar que permite el soporte de calidad de servicio en las redes WLAN existentes (IEEE 802.11a/b/g), el estándar IEEE 802.11e.

Durante su desarrollo, se ha realizado la implementación de un planificador de paquetes utilizado por el mecanismo HCCA (acceso al medio centralizado que permite garantizar el cumplimiento de requisitos de QoS, incluido en el estándar IEEE 802.11e). Este algoritmo de planificación ha sido elegido, entre los existentes en la literatura, según criterios de mayor eficiencia en la gestión de tráfico con distintas características, menor complejidad computacional y facilidad de implementación.

El algoritmo seleccionado, *Wireless Timed Token Protocol (WTTT)*, ha sido implementado y posteriormente analizado mediante simulación utilizando la herramienta *Network Simulator 2*, simulador de licencia gratuita y de código abierto, ampliamente utilizado por la comunidad científica.

De este análisis se puede concluir que el planificador WTTT garantiza la tasa de bit negociada para flujos de tráfico tanto CBR (tasa constante) como VBR (tasa variable) en los escenarios simulados. Además, este algoritmo es capaz de servir flujos de tráfico VBR con un retardo máximo casi constante respecto al número de flujos de tráfico admitidos. Además, se ha comprobado que el retardo permanece estable en un escenario de tráfico con fuentes CBR y VBR. Finalmente, se analizan otros aspectos como la variación del retardo (*jitter*) y el tiempo entre dos transmisiones consecutivas de una misma fuente.

RESOURCE MANAGEMENT FOR REAL TIME SERVICES IN WIRELESS LOCAL AREA NETWORKS

(José Antonio Rodríguez Salas)

KEYWORDS

Enhanced Distributed Channel Access (EDCA), HCF Controlled Channel Access (HCCA), Hybrid Coordinator (HC), Network Simulator 2 (*NS2*), Quality of Service (QoS), Scheduler, Wireless Local Area Network (WLAN), Wireless Timed Token Protocol (WTTTP).

ABSTRACT

This project focuses on the analysis and improvement of current WLAN networks to support the Quality of Service required by home entertainment services, such as voice calls and video on demand. In order to achieve this goal, the IEEE 802.11e standard has been designed to provide QoS guarantees on current WLAN networks (IEEE 802.11a/b/g).

A packet scheduling has been implemented using the HCCA mechanism (centralized medium access mechanism that guarantees QoS requirements, included in the IEEE 802.11e standard). This algorithm has been chosen, among those in the current literature, based on its efficiency for traffic scheduling with different services, its computational complexity, and ease of implementation.

The Wireless Timed Token Protocol (WTTTP) algorithm has been implemented and evaluated by means of Network Simulator 2, a free license and open source simulator, which is widely used by the scientific community.

This analysis concludes that the WTTTP scheduler guarantees the negotiated data rate for both CBR (constant bit rate) and VBR (variable bit rate) flows in the selected scenarios. In addition, the maximum delay remains almost constant regarding the number of admitted traffic flows. The average delay is steady in a simulated scenario with both CBR and VBR sources. Finally, other relevant metrics, such as delay variation (jitter) and the time between packet arrivals from the same source, have been analyzed.

D. Jorge Navarro Ortiz

Profesor del departamento de ***Teoría de la señal, Telemática y Comunicaciones***
de la Universidad de Granada, como director del Proyecto Fin de Carrera de
D. ***José Antonio Rodríguez Salas***

Informa:

que el presente trabajo, titulado:

***GESTIÓN DE RECURSOS PARA SERVICIOS DE TIEMPO REAL
SOBRE REDES WLAN***

Ha sido realizado y redactado por el/la mencionado/a alumno/a bajo nuestra
dirección, y con esta fecha autorizo a su presentación.

Granada, a _____ de _____ de 20____

Fdo. _____

Los abajo firmantes autorizan a que la presente copia de Proyecto Fin de Carrera se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, a _____ de _____ de 20____

(Firmas y números de DNI del alumno y de los tutores)

Agradecimientos

Tras varios meses de intenso trabajo, finalmente ha podido ver la luz mi proyecto fin de carrera. Mientras ha durado este esfuerzo, han estado siempre a mi lado las personas que a continuación citaré y con las que tengo contraída una deuda clara que nunca agradeceré bastante.

En primer lugar, quiero dar las gracias a mi tutor de proyecto D. Jorge Navarro Ortiz, por su ayuda y paciencia durante todo este tiempo. Sus consejos acertados en todo momento, me han hecho superar todos los escollos que han surgido.

En segundo lugar, también quisiera dar las gracias sinceras a mis compañeros y amigos, por todos los momentos que hemos pasado juntos, por sus ánimos constantes y el cariño que me han mostrado siempre.

Muy importante ha sido el apoyo incondicional y la confianza de mis padres. Sin sus atenciones, nada de esto hubiera sido posible

Por último, invitar a cuantos puedan estar interesados por el mundo de las redes inalámbricas y su uso para servicios en tiempo real, que no duden en mirar estas páginas, que he hecho con la mayor ilusión y atención.

José Antonio Rodríguez Salas

Julio 2009.

Índice

Índice de figuras y tablas.....	12
Glosario de términos.....	16
1. Introducción.....	27
2. IEEE 802.11.....	29
2.1 Componentes de IEEE 802.11.....	29
2.1.1 Estaciones.....	30
2.1.2 Medio inalámbrico.....	30
2.1.3 Puntos de acceso.....	30
2.1.4 Sistema de distribución.....	31
2.2 Tipos de redes.....	31
2.2.1 Conjunto de servicios básicos.....	31
2.2.1.1 Redes independientes.....	31
2.2.1.2 Redes de infraestructura.....	32
2.2.2 Conjunto de servicios extendidos.....	33
2.3 Arquitectura de capas.....	34
2.3.1 Capa física.....	35
2.3.2 Capa de enlace	37
2.3.3 Capa MAC IEEE 802.11.....	38
2.3.3.1 Función de coordinación distribuida.....	39
2.3.3.2 Función de coordinación puntual.....	42
2.3.3.3 Espacio entre tramas.....	50
2.3.3.4 Conclusión sobre PCF y DCF.....	51
3. IEEE 802.11e.....	52
3.1 Calidad de servicio.....	54
3.2 Capa MAC IEEE 802.11e.....	58
3.2.1 EDCA.....	59
3.2.2 HCCA.....	63
3.3 Algoritmos para IEEE 802.11e.....	67

3.3.1 Algoritmos basados en el tiempo de expiración de paquete.....	79
3.3.1.1 RTH.....	70
3.3.1.2 SETT-EDD.....	71
3.3.1.3 Algoritmo de Korakis y Tassiulis.....	72
3.3.2 Algoritmos basados en estimación de la longitud de la cola.....	73
3.3.3 Algoritmos híbridos HCCA/EDCA.....	73
3.4 Conclusión.....	73
4. WTPP.....	75
4.1 Gestión de la cola con disciplina <i>Round Robin</i>	76
4.2 Cálculo del tiempo de transmisión.....	77
4.3 Control de admisión.....	78
4.4 Cálculo de los parámetros $TTRT$ y H_i	79
5. Network Simulator 2.....	85
5.1 Características generales.....	85
5.1.1 Ejemplo de script de simulación.....	87
5.1.2 WLAN en NS2.....	90
5.2 Módulos añadidos.....	91
5.3 Modificación de NS2: implementación de WTPP.....	94
6. Simulaciones y resultados.....	95
6.1 Consideraciones iniciales.....	95
6.2 Escenarios y resultados.....	97
6.2.1 Escenario1.....	97
6.2.2 Escenario2.....	101
6.2.3 Escenario3.....	104
7. Conclusiones.....	107

8. Referencias.....	108
Anexo I .- Script OTcl.....	109
Anexo II .- Archivo awk.....	116
Anexo III.- Archivo Matlab: ASCII → binario.....	119

Índice de figuras y tablas

Índice de figuras

2. IEEE 802.11

Figura 2.1.- Componentes de una red inalámbrica IEEE 802.11

Figura 2.2.- Ejemplo de red independiente

Figura 2.3.- Ejemplo de ESS

Figura 2.4.- Arquitectura TCP/IP

Figura 2.5.- Capa física del estándar IEEE 802.11

Figura 2.6.- Capa de enlace IEEE 802.11

Figura 2.7.- Funcionamiento modo básico

Figura 2.8.- Funcionamiento modo RTS/CTS

Figura 2.9.- Topología red inalámbrica

Figura 2.10.- Ejemplo IEEE 802.11

Figura 2.11.- Algoritmo PCF

Figura 2.12.-Trama CF-POLL

Figura 2.13.-Trama DATOS+CF-ACK

Figura 2.14.- Trama CF-ACK+CF-POLL

Figura 2.15.- Trama DATOS+CF-ACK+CF-POLL

Figura 2.16.- Supertrama IEEE 802.11e

Figura 2.17.- Espacio entre tramas IEEE 802.11

3. IEEE 802.11e

Figura 3.1.- Esquema IEEE 802.11e

Figura 3.2.- Funciones de acceso al canal IEEE 802.11 y IEEE 802.11e

Figura 3.3.- Tráfico EDCA

Figura 3.4.- Parámetros EDCA

Figura 3.5.- Tráfico HCCA

Figura 3.6.- Supertrama IEEE 802.11e

Figura 3.7.- Reparto de las TXOPs del algoritmo básico de IEEE 802.11e

Figura 3.8.- Intervalos de servicio del algoritmo IEEE 802.11e

Figura 3.9.- Elementos del planificador RTH

Figura 3.10.- Ejemplo RTH

4. WTPP

Figura 4.1.- Gestión cola con disciplina *round robin*

Figura 4.2.- Algoritmo WTPP

5. Network Simulator 2

Figura 5.1.- Dualidad entre clases

Figura 5.2.- Vista simplificada de los elementos de NS2

Figura 5.3.- Estructura de directorios de NS2

Figura 5.4.- Escenario para un script simple de simulación

Figura 5.5.- Capa MAC en NS2

Figura 5.6.- Diagrama de clases HCCA

Figura 5.7.- Esquema de VoIP para NS2

6. Simulaciones y resultados6.7.

Figura 6.1.- Retardo (tanto del percentil 99 como del promedio) e intervalo de *polling* del enlace ascendente con un flujo VBR.

Figura 6.2.- *Throughput* del enlace ascendente con tráfico VBR

Figura 6.3.- Tiempo EDCA del enlace ascendente con tráfico VBR

Figura 6.4.- CDF del retardo del enlace ascendente con tráfico VBR

Figura 6.5.- Percentil 99 del retardo para varios valores del TTRT

Figura 6.6.- Retardo medio para varios valores del TTRT

Figura 6.7.- *Null ratio* del enlace ascendente con tráfico VBR

Figura 6.8.- Intervalo de *polling* promedio de los flujos ascendentes con tráfico CBR y VBR bidireccional

Figura 6.9.- Percentil 99 del retardo para los enlaces ascendentes y descendentes con tráfico CBR y VBR bidireccional.

Figura 6.10.- *Throughput* medio para los enlaces ascendente y descendente con tráfico VBR y CBR

Figura 6.11.- Tiempo EDCA para el escenario 1 y 3

Índice de tablas

2. IEEE 802.11

Tabla 2.1.- Resumen de los estándares IEEE 802.11

Tabla 2.2.- Valores de los espacios entre tramas IEEE 802.11

3. IEEE 802.11e

Tabla 3.1.- Mapa de prioridades de usuario a categorías de acceso (AC)

Tabla 3.2.- Configuración estándar para IEEE 802.11b

Tabla 3.3.- Configuración del estándar para IEEE 802.11g

Tabla 3.4.- Configuración del estándar para IEEE 802.11e

6. Simulaciones y resultados

Tabla 6.1.- Parámetros de la capa MAC y de la capa física

Tabla 6.2.- Formato de las trazas de salida de NS2

Glosario de términos

A

AC (Categoría de Acceso)

Valor que indica la prioridad del tráfico cuando se utiliza el mecanismo EDCA. Existe una cola por cada categoría de acceso.

ACK (Trama de Confirmación)

Es un mensaje que se envía para confirmar que un mensaje ha llegado correctamente a su destino.

AIFS (Espacio entre Tramas Arbitrario)

En el modo EDCA, tras una ranura de tiempo ocupada, tiempo que debe esperar una estación hasta poder decrementar su contador de *backoff* y que depende de la cola de tráfico en cuestión.

AP (Punto de Acceso)

Dispositivo que realiza la función de conectar a las estaciones.

B

BS (Tamaño de la Ráfaga)

Parámetro de la especificación del tráfico que nos indica el máximo tamaño de ráfagas de datos de un determinado flujo de datos.

BSS (Conjunto de Servicio Básico)

Son los servicios que soportan la comunicación entre estaciones.

BSA (Área de Servicio Básico)

Área dentro de la cuál una estación puede comunicarse con otras estaciones, que también pertenecen al área de servicio básico.

C

CAP (Período de Acceso Controlado)

Un período de acceso controlado consiste en una o más TXOPs, durante las cuales el QAP puede transmitir SDUs que pertenecen a un flujo descendente o enviar un mensaje de *polling* a una o más estaciones, especificando el tiempo máximo durante el que pueden ocupar el medio. Todo esto dentro de un intervalo de contención.

CBR (Tasa de Bit Constante)

Tráfico de usuario caracterizado por una velocidad de transmisión constante.

CCK (Modulación de Código Complementario)

Esquema de modulación usado en redes inalámbricas que emplean el estándar IEEE 802.11b.

CDF (Función de Densidad de Probabilidad Acumulada)

Nos indica la probabilidad de que la variable aleatoria sea menor a un valor x

CFP (Período Libre de Contención)

Durante el acceso al canal en IEEE 802.11, período en el que el coordinador puntual controla quién transmitirá, eliminando la necesidad de que las estaciones compitan por conseguir el acceso al canal.

CP (Período de Contención)

Durante el acceso al canal en IEEE 802.11, período en el cual las estaciones compiten por el canal.

CRC (Código de Redundancia Cíclica)

Campo usado como comprobación para detectar la alteración de datos durante su transmisión.

CSMA/CA (Acceso Múltiple con Detección de Portadora Evitando Colisiones)

Protocolo de control de acceso al medio que permite que varias estaciones utilicen un mismo medio de transmisión. Este mecanismo no detecta colisiones pero sí intenta minimizarlas.

CSMA/CD (Acceso Múltiple por Detección de Portadora con Detección de Colisiones)

Protocolo de control de acceso al medio en el que cada estación que desea transmitir, previamente escucha el canal, detectando posibles colisiones.

CTS (Clear To Send)

En el modo RTS/CTS, este mensaje indica a una estación que puede transmitir.

CW (Ventana de Contención)

Parámetro del estándar IEEE 802.11 que corresponde al valor máximo posible del tiempo aleatorio de *backoff*.

D

D (Retardo)

Variable de la especificación del tráfico que define el tiempo máximo para el transporte de tramas, incluyendo la propagación por el medio inalámbrico y el retardo en la cola.

DCF (Función de Coordinación Distribuida)

Función de la capa MAC que permite la transmisión de datos de acuerdo con el método de mejor esfuerzo. Con DCF, todas las estaciones compiten por conseguir el canal cada vez que intentan transmitir.

DIFS (Espacio entre tramas DCF)

Tiempo mínimo que el medio ha de estar sin ocupar para poder transmitir durante el período de contención.

DSSS (Espectro Ensanchado por Secuencia Directa)

Método de modulación en espectro ensanchado para transmisión de señales digitales. Esta modulación utiliza un código pseudoaleatorio para modular la señal a transmitir.

E

EDCA (Acceso Coordinado y Distribuido Mejorado)

Función de acceso al canal de IEEE 802.11e basada en contención, y que constituye una ampliación de la función DCF.

EDF (El Primero en Servirse el de la Fecha de Cumplimiento más Temprana)

Algoritmo que asigna mayor prioridad a las tareas con menor tiempo de servicio.

EIFS (Espacio Entre Tramas Extendido)

Tiempo entre tramas usado para retransmisiones, siendo el espacio entre trama de mayor duración.

ESS (Conjunto de Servicios Extendidos)

Conjunto de servicios que permiten la unión de varias BSSs mediante un sistema de distribución

F

FDDI (Interfaz de Datos Distribuidos por Fibra)

Estándar para la transmisión de datos en redes de computadoras de área extendida o de área local mediante cable de fibra óptica.

FDBS (Planificador Dinámico Basado en Realimentación)

Algoritmo de planificación basado en estimación de la longitud de la cola.

FHCF (HCF Justo)

Algoritmo de planificación basado en estimación de la longitud de la cola.

FHSS (Espectro Ensanchado por Salto de Frecuencia)

Es una técnica de modulación en espectro ensanchado en el que la señal se emite sobre una serie de frecuencias pseudoaleatorias, saltando de frecuencia en frecuencia síncronamente con el receptor.

FTP (Protocolo de Transferencia de archivos)

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

H

HCCA (Mecanismo de Acceso al Medio Controlado por HCF)

Es una ampliación de PCF para proveer QoS. Se puede usar tanto en el intervalo de contención como en el intervalo libre de contención. El acceso al canal está basado en el envío de mensajes de *polling*, siendo una función centralizada.

HCF (Función de Coordinación Híbrida)

Es la función encargada de controlar el acceso al medio en el estándar IEEE802.11e.

HC (Coordinador Híbrido)

En HCCA, es un dispositivo encargado de decidir a qué estación le toca transmitir. Normalmente coincide con el QAP.

H_i (Ancho de Banda Síncrono)

Tiempo fijo asignado por el planificador WTTT en cada ciclo

I

IBSS (Conjunto de Servicio Básico Independiente)

Conjunto de servicios que soportan la comunicación entre estaciones, sin usar un punto de acceso.

IP (Protocolo de Internet)

Protocolo que corresponde con la capa de red de la arquitectura TCP/IP, y proporciona a la capa de transporte un servicio no orientado a conexión y de entrega de mejor esfuerzo.

IR (Infrarrojo)

Radiación electromagnética con longitudes de onda entre 700 nanómetros hasta 1 milímetro.

ISM (Industrial, Científico y Médicas)

Son bandas reservadas internacionalmente para un uso no comercial de radiofrecuencias electromagnéticas en las áreas industrial, científica y médica.

L

LAN (Red de Área Local)

Está formado por un conjunto de ordenadores y dispositivos de red que están interconectados mediante un medio de transmisión compartido.

LLC (Control del Enlace Lógico)

Subcapa de la capa de enlace que se encarga del control de errores, el control de flujo y el entramado.

M

M (Tamaño Máximo de la SDU)

Parámetro de la especificación del tráfico que define el tamaño máximo de la SDU.

MAC (Control de Acceso al Medio)

Subcapa de la capa de enlace que se encarga de coordinar el acceso al canal de forma que la información vaya desde el origen al destino a través de la misma red de difusión.

MACAW (*Multiple Access with Collision Avoidance for Wireless*)

Algoritmo de acceso al canal para el modo RTS/CTS.

Msi (Mínimo Intervalo de Servicio)

Parámetro del TSPEC que indica el intervalo de tiempo mínimo entre el comienzo de períodos de servicio sucesivos.

MSI (Máximo Intervalo de Servicio)

Parámetro del TSPEC que indica el intervalo de tiempo máximo entre el comienzo de períodos de servicio sucesivos.

N

N (Tamaño Nominal de la SDU)

Variable del TSPEC que define el tamaño promedio de la SDU

NAM (Animador de Red)

Simulador gráfico incorporado en NS2

NAV (Vector de Asignación de red)

Este vector recuerda a una estación que no puede transmitir su propio tráfico durante un tiempo y de esta forma poder realizar transmisiones procedentes de otras estaciones a través de él.

NS2 (Network Simulator 2)

Simulador de redes de eventos discretos y orientado a objetos (Java, C++,...) desarrollado por la Universidad de Berkeley en California y escrito en C++ y OTcl.

O

OFDM (Multiplexación por División de Frecuencias Ortogonales)

Es una técnica de modulación que consiste en enviar un conjunto de portadoras de diferentes frecuencias donde cada una transporta información modulada mediante otra técnica.

P

PBCC (Codificación Convolutiva de Paquete Binario)

Esquema de modulación usado en redes inalámbricas que emplean el estándar IEEE 802.11b.

PC (Coordinador puntual)

En la función PCF, el coordinador puntual es el encargado de decidir qué estación tiene el turno para poder transmitir.

PCF (Función de Coordinación Puntual)

Función centralizada de la capa MAC. Utiliza un coordinador puntual para repartir a las estaciones los turnos de transmisión.

PIFS (Espacio Entre Tramas PCF)

Tiempo de espera antes de poder transmitir una trama durante el período libre contención. Como el intervalo PIFS es menor que el período DIFS, estas transmisiones tendrán prioridad durante dicho período.

PMD (Capa Dependiente del Medio Físico)

Subcapa encargada de la modulación y codificación dentro de la capa física.

PLCP (Procedimiento de Convergencia de la Capa Física)

Subcapa de la capa física encargada de tratar las tramas generadas en la subcapa MAC y adaptarlas al formato adecuado para la subcapa dependiente del medio físico.

Q

QAP (Punto de Acceso con QoS)

Punto de acceso con soporte de calidad de servicio.

QoS (Calidad de Servicio)

Parámetro multidimensional que caracteriza lo que demanda la aplicación y/o lo que ofrece la red.

QSTA (Estación con QoS)

Estación con soporte de calidad de servicio.

QBSS (Conjunto de Servicio Básico con QoS)

Conjunto de servicio básico con soporte de calidad de servicio.

R

R (Tasa Mínima en la Capa Física)

Parámetro de la especificación del tráfico que define la tasa de bit mínima en el medio inalámbrico.

ρ (Tasa de Datos Promedio)

Parámetro de la especificación del tráfico que define la tasa de bit promedio.

RTH (HCCA para Tiempo Real)

Pertenece al conjunto de algoritmos de planificación basado en el tiempo de expiración del paquete.

RTS (Request To Send)

En el modo RTS/CTS, este mensaje indica que una estación quiere transmitir.

S

SETT-EDD (Planificación Basada en Tiempos Estimados de Transmisión– Fecha de Cumplimiento más Temprana)

Es un algoritmo de planificación incluido en el grupo de los algoritmos basados en el tiempo límite para transmitir los paquetes.

SDU (Unidad de Datos de Servicio)

Es el conjunto de datos, dentro del sistema de referencia OSI que, dada una capa concreta, no incluyen la información de control o cabecera de dicha capa.

SIFS (Espacio entre tramas corto)

Tiempo de espera antes de realizar transmisiones de alta prioridad (e.g. tramas ACK). También se utiliza para transmisiones de las estaciones durante el período libre de contención. Debido a su corta duración, este tipo de transmisión no presenta colisiones.

SRP (Protocolo de Pila de Recursos)

Algoritmo usado en RTH para poder bloquear las tareas, y poder llevar a cabo la planificación de los recursos con EDF.

SSID (Identificador de Conjunto de Servicios)

Es un código incluido en todos los paquetes de una red inalámbrica para identificarla. El código consiste en un máximo de 32 caracteres alfanuméricos. Todos los dispositivos inalámbricos que intentan comunicarse entre sí tienen que compartir el mismo SSID. A menudo al SSID se le conoce como nombre de red.

SST (Tiempo de Comienzo del Servicio)

Parámetro del TSPEC que nos indica el momento en el que el intervalo de servicio comienza.

STA (Estación)

Acrónimo asignado a los equipos terminales de usuario.

T

TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet)

Es un conjunto de protocolos en el que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. Hace referencia a los dos protocolos más importantes que lo componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

TCP (Protocolo de Control de Transmisión)

Protocolo perteneciente a la capa de transporte de la arquitectura TCP/IP. Proporciona un servicio con control de flujo, de congestión y de errores.

TGe (Task Group E)

Grupo encargado de diseñar el estándar IEEE 802.11e.

TID (Identificador de Tráfico)

Este identificador tomará un valor entre 0 y 15, según sus requisitos de QoS. Las tramas con valores TID entre 0 y 7 son mapeadas en cuatro colas, correspondientes a cuatro categorías de acceso (ACs) y usan reglas de acceso EDCA. Por el contrario, las tramas marcadas con un TID entre 8 y 15 son mapeadas en ocho tipos de flujo de tráfico (TS) y usan reglas HCCA.

TTRT (Target Token Rotation Time)

En el algoritmo WTTT, este parámetro indica el tiempo de duración de un ciclo completo de la lista de turnos.

Token Rotation Timer (TRT_i)

Temporizador del algoritmo WTTT propio de cada nodo. Mide el tiempo que ha pasado desde la última vez que el nodo recibió un turno para poder transmitir.

TS (Flujo de Tráfico)

En IEEE 802.11e define un flujo de datos, ascendente y descendente, con calidad de servicio.

TSID (Identificación del tipo de flujo)

Parámetro del TSPEC para identificar cada uno de los flujos de una estación.

TSPEC (Especificación del Tráfico)

Conjunto de parámetros que definen las características del tráfico de una determinada estación.

Target Beacon Transmisión Time (TBTT)

Parámetro de la trama *beacon* que indica el tiempo entre transmisión de dos tramas *beacon*.

TXOP (Oportunidad de Transmisión)

Es un intervalo de tiempo limitado durante el cual una QSTA puede transmitir una serie de tramas.

U

UDP (Protocolo de Datagramas de Usuario)

Es un protocolo de nivel de transporte que permite el envío de información a través de la red sin que se haya establecido previamente una conexión, existan confirmaciones ni control de flujo o congestión.

UN-II (Infraestructura de Información Nacional sin Licencia)

Banda del espectro radiofrecuencia entre 5,15-5,825GHz.

V

VBR (Tasa de Bit Variable)

Tráfico con variaciones ocasionales del caudal, manifestándose éstas en forma de ráfagas.

VC (Videoconferencia)

Tipo de aplicación de video bajo demanda, en la que dos o varias personas situadas de forma remota, se comunican mediante video y audio.

VoIP (Voz sobre IP)

Tecnología que hace posible que la señal de voz viaje a través de redes de datos empleando el protocolo IP (Protocolo Internet). Esto significa que se envía la señal de voz en forma digital en paquetes en lugar de enviarla (en forma digital o analógica) a través de circuitos utilizables sólo para telefonía.

VoD (Video Bajo Demanda)

Es un protocolo que permite a los usuarios acceder a contenidos multimedia de forma personalizada.

W

WLAN (Redes de Área Local Inalámbrica)

Redes de área local no cableadas. Existen varios tipos de tecnologías WLAN, como pueden ser IEEE 802.11 o HiperLAN.

WDS (Sistema de Distribución Inalámbrica)

Un sistema de distribución inalámbrico (WDS) es un sistema que permite la interconexión inalámbrica de puntos de acceso en una red IEEE 802.11. Permite que una red inalámbrica pueda ser ampliada mediante múltiples puntos de acceso sin la necesidad de un cable troncal que los conecte.

WTPP (Wireless Timed Token Protocol)

Algoritmo de planificación para el mecanismo de acceso al medio HCCA del estándar IEEE 802.11e.

1. Introducción

En los últimos años, el despliegue de tecnologías inalámbricas ha tenido un crecimiento exponencial. El desarrollo de la sociedad de la información y la comunicación ha contribuido a ello. Entre las nuevas tecnologías desplegadas, sobresalen las comunicaciones mediante teléfonos móviles, ordenadores portátiles y otros dispositivos de comunicación inalámbrica.

Además, se ha producido un aumento del número de aplicaciones multimedia, tales como voz sobre IP (VoIP), la audio conferencia, la video conferencia, el *streaming* de audio y el *streaming* de video. Estas aplicaciones requieren que los servicios suministrados a los usuarios tengan calidad de servicio (QoS). En concreto, muchas aplicaciones multimedia son sensibles a la tasa de transmisión, al retardo extremo a extremo y a las variaciones del retardo (*jitter*), aunque toleran cierta pérdida de datos. Por otro lado, las aplicaciones flexibles al retardo y al *jitter*, como pueden ser el correo electrónico, la navegación web y el acceso remoto, tienen requisitos distintos a las aplicaciones multimedia. En estas aplicaciones los retardos elevados, aunque no son deseados, no impiden que se realice el servicio.

Las aplicaciones comentadas utilizan IP como protocolo de capa de red. Este protocolo proporciona un servicio de mejor esfuerzo (*best-effort*) para todos los datos enviados, sin diferenciar el tipo de tráfico y sus requisitos de QoS, es decir, ofrece un servicio de interconexión de redes independiente de los requisitos del usuario, con un tratamiento *igualitario* para todos los datagramas IP.

Por otro lado, IEEE 802.11 se ha convertido en el estándar más usado para redes WLAN. Este proyecto se centra en el estudio de las redes WLAN para soportar la calidad de servicio requerida por los servicios típicos de entretenimiento en el hogar, telefonía, vídeo bajo demanda, servicios típicos de Internet,...

En la actualidad existe un estándar que permite mejorar la calidad de servicio en las redes WLAN existentes (802.11a/b/g), el estándar 802.11e. Para cumplir con este objetivo, IEEE 802.11e introduce un nuevo elemento llamado Función de Coordinación Híbrida (HCF) con dos tipos de acceso:

- Acceso Coordinado y Distribuido Mejorado (EDCA)
- Acceso al Canal Controlado por HCF (HCCA)

EDCA no garantiza que se cumplan los requisitos de QoS estrictamente, solamente da prioridades a los distintos tipos de tráfico. Sin embargo, en las aplicaciones descritas anteriormente, se puede hacer necesario garantizar un retardo y un *jitter* limitado. Para conseguir esto, se diseñó el mecanismo de acceso al canal controlado por HCF (HCCA), el cual garantiza unos requisitos de QoS para el tráfico.

En este proyecto, se realizará un estudio de los planificadores existentes en la literatura para funcionar sobre IEEE 802.11e y proveer calidad de servicio con los

requisitos garantizados. Seguidamente, se verán las ventajas y desventajas de cada uno de ellos y se seleccionará uno de los algoritmos de planificación según criterios de mayor eficiencia, menor carga computacional y que se adapte mejor a los distintos tipos de tráfico generados por estas aplicaciones. Por último, se simulará este planificador de paquetes que será utilizado por el mecanismo HCCA (mecanismo de acceso al medio controlado HCF). Esta implementación se llevará a cabo en *Network Simulator 2*, simulador de licencia gratuita y de código abierto, ampliamente utilizado por la comunidad científica.

Por tanto, el objetivo principal de este proyecto, será la implementación de un algoritmo planificador que, operando sobre IP, satisfaga las demandas requeridas por las aplicaciones en redes IEEE 802.11. Posteriormente, se evaluará su rendimiento con diferentes patrones de tráfico y se comprobará así que se garantizan los requisitos de QoS a todas las estaciones admitidas por el planificador.

2. IEEE 802.11

Las redes de área local inalámbricas (WLAN) son una alternativa a las redes de área local cableadas. Esta tecnología utiliza ondas electromagnéticas (radio o infrarrojos) para transmitir datos, eliminando la necesidad de cables. Sin embargo, WLAN presenta una serie de inconvenientes frente a las redes cableadas, como tener una tasa de transmisión menor y una interfaz radio con condiciones cambiantes, que pueden afectar a la transmisión de datos a través del medio inalámbrico (e.g. aumentando la probabilidad de error).

El estándar más utilizado de WLAN es IEEE 802.11, al que se le han ido añadiendo nuevos estándares (e.g. IEEE 802.11a/b/g/e/i) a éste, cuyos objetivos principales se exponen a continuación:

- Aumento de la tasa de transmisión: las redes inalámbricas tienen un ancho de banda limitado y son más sensibles a factores ambientales o externos (obstáculos, interferencias,...) que las redes cableadas. Estos factores disminuyen la tasa de transmisión, de tal forma que se hace necesaria la inclusión de mecanismos que permitan la adaptación a las condiciones del canal.
- Mejorar la seguridad: controlar el acceso al medio inalámbrico resulta más complicado que acceder al medio en las redes cableadas, ya que cualquier usuario con una tarjeta de red inalámbrica puede acceder a él. Por tanto, hay que dotar a las redes WLAN de mecanismos de seguridad para evitar que un usuario no autorizado pueda entrar a nuestra red.
- Garantizar los requisitos de QoS: se tienen que cumplir los requisitos de QoS demandados por las aplicaciones. Como resultado, el tráfico será tratado de forma diferente según sus requisitos de calidad de servicio.
- Controlar el acceso al medio: hay que gestionar el acceso al medio, de forma que los distintos usuarios conectados a la red lo puedan compartir de manera eficiente.

A continuación se detallan los componentes de una red IEEE 802.11. Posteriormente, en el apartado 2.2 se explican los distintos tipos de redes que se pueden formar uniendo los elementos de IEEE 802.11. Por último, en el apartado 2.3 se describe la arquitectura IEEE 802.11, tomando como punto central la capa MAC, donde se encuentran las funciones DCF y PCF.

2.1 Componentes de IEEE 802.11

Como se puede observar en la figura 2.1, IEEE 802.11 tiene cuatro componentes principales:

1. Estaciones
2. Medio inalámbrico

3. Punto de acceso
4. Sistema de distribución

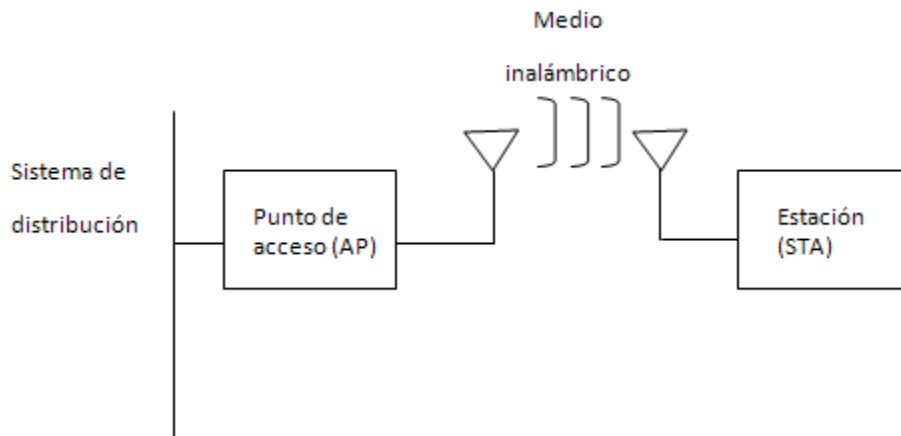


FIG 2.1: COMPONENTES DE UNA RED INALÁMBRICA IEEE 802.11

2.1.1 Estaciones

Las estaciones (STAs) son dispositivos que actúan como origen y destino de los datos transmitidos. El objetivo de las redes inalámbricas es permitir la transmisión de datos entre estaciones. Como se verá posteriormente, para el caso de IEEE 802.11e las denominaremos QSTAs ya que soportan calidad de servicio.

2.1.2 Medio inalámbrico

El medio inalámbrico es el soporte que permite la transferencia de datos entre estaciones. El estándar permite dos tecnologías diferentes para la propagación de la señal, radiofrecuencia e infrarrojos, siendo ésta última la más utilizada.

2.1.3 Puntos de Acceso

Un Punto de Acceso (AP) es una estación que permite conectar otras estaciones al sistema de distribución. Los puntos de acceso se sitúan de forma que puedan proporcionar la cobertura necesaria para dar servicio a los terminales que no tienen comunicación directa, aumentando su radio de cobertura.

Además, el AP centraliza todas las comunicaciones entre STA, ya que, si dos estaciones quieren comunicarse entre sí, deben hacerlo a través del AP. Por tanto, el radio de cobertura de un AP limita la distancia a la cuál puede comunicarse una determinada estación. Sin embargo, es posible aumentar la cobertura de la red mediante un sistema de distribución, punto que se abordará a continuación.

2.1.4 Sistema de distribución

Un sistema de distribución está formado por varios puntos de acceso conectados entre sí mediante alguna tecnología, de forma que se pueda obtener un área de cobertura mayor. Los puntos de acceso deben comunicarse para gestionar la movilidad de las estaciones. La tecnología más habitual en los sistemas de distribución es Ethernet, aunque se pueden utilizar otras tecnologías, incluso el estándar IEEE 802.11 creando un Sistema de Distribución Inalámbrico (WDS).

Cuando una estación móvil se mueve de una zona de cobertura de un AP a la de otro (*roaming*), se hace evitando los cortes en la comunicación y la pérdida de cobertura.

2.2 Tipos de redes

Con los elementos descritos anteriormente se pueden formar distintos tipos de redes. Para poder crearlas, el estándar IEEE 802.11 define dos conjuntos de servicios que son el Conjunto de Servicio Básico (BSS) y el Conjunto de Servicio Extendido (ESS).

2.2.1 Conjunto de servicios básicos

Un Conjunto de Servicios Básicos (BSS) consisten en los servicios que soportan la comunicación entre STAs. Estas comunicaciones tienen lugar dentro de un área llamada Área de Servicios Básicos (BSA), que dependerá de las características del medio inalámbrico. Cuando una estación está dentro de un BSA puede comunicarse con los otros miembros del área de servicios básicos.

Los BSS dan soporte a las redes independientes y a las redes de infraestructura, descritas a continuación.

2.2.1.1 Redes Independientes

Las estaciones en un BSS independiente (IBSS) se comunican directamente unas con otras. Típicamente, las IBSSs están compuestas por un pequeño número de estaciones preparadas para un propósito específico con conectividad durante un período corto de tiempo. Un uso habitual es la creación de una red para soportar una reunión en una sala de conferencias. Cuando la reunión comienza, los participantes crean un IBSS para compartir datos. Al final de la reunión, se cancela la red. Debido a su corta duración, pequeño tamaño y propósito específico, a las IBSSs también se les llama *ad-hoc* BSS o redes *ad-hoc*. En la figura 2.2, se muestra un ejemplo de una red independiente.

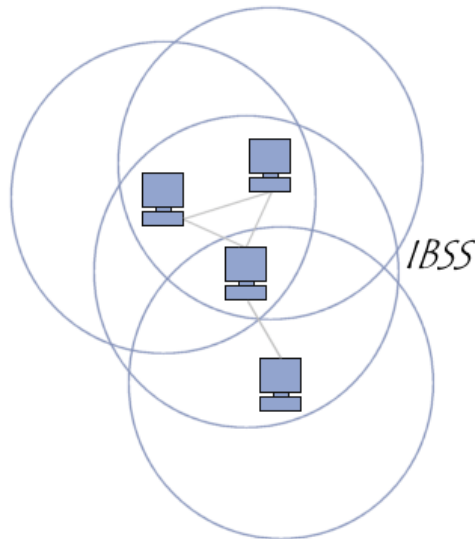


FIG2.2: EJEMPLO DE RED INDEPENDIENTE

2.2.1.2 Redes de infraestructura

Las redes de infraestructura se distinguen de las redes *ad-hoc* por el uso de un punto de acceso. Los puntos de acceso centralizan todas las comunicaciones entre las distintas estaciones, incluyendo comunicaciones entre estaciones en la misma área de servicio básico y comunicaciones entre estaciones con el sistema de distribución. De esta forma, la estación origen transfiere una trama al punto de acceso, y éste la transmite a la estación destino, a través del sistema de distribución. Por tanto, no es necesario que dos STAs estén ambas en su radio de cobertura para poder comunicarse, sino que es suficiente con que estén dentro del radio de cobertura del AP.

El área de servicio básico está limitado por el espacio en el que las transmisiones, desde el punto de acceso, pueden ser recibidas por las estaciones. La transmisión descrita anteriormente requiere más capacidad para transmitir que una trama directa entre emisor y receptor como ocurría en redes *ad-hoc*, ya que se duplica el envío de tramas. Primero se envían las tramas de la estación origen al AP y posteriormente se envían del punto de acceso a la estación destino.

En las redes de infraestructura, las estaciones deben asociarse con un punto de acceso para poder transmitir datos. La asociación es el proceso por el cual las estaciones se unen a una red IEEE 802.11. No es un proceso simétrico, ya que las estaciones siempre inician el proceso de asociación y el AP sólo puede elegir si concede o deniega el acceso. Una estación sólo puede asociarse a un único AP, aunque no hay ningún límite para el número de estaciones que puede soportar un AP. Sin embargo, aumentar el número de estaciones puede producir una disminución de la capacidad asignada a cada una de ellas, por lo que existe un límite práctico para el número de estaciones asociadas.

Como resumen, algunas de las ventajas e inconvenientes de las redes de infraestructura son:

- Todas las estaciones móviles tienen que estar dentro del alcance del punto de acceso, pero no hay restricciones entre la distancia entre ellas, como ocurre en las redes ad-hoc.
- Gracias al proceso de asociación se consigue un control de acceso que no existía en las redes *ad-hoc*.
- La falta de comunicación directa entre estaciones reduce la capacidad de transmisión en las redes de infraestructura.
- Las redes *ad-hoc* son más sencillas de configurar y no necesitan puntos de acceso, por lo que requieren menos recursos que las redes de infraestructura.

2.2.2 Conjunto de Servicios Extendidos

El conjunto de servicios básicos pueden dar cobertura a una pequeña oficina o casa, pero no pueden dar servicio a zonas grandes. El estándar IEEE 802.11 permite que las redes inalámbricas cubran áreas mayores mediante la unión de BSSs formando un Conjunto de Servicios Extendidos (ESS).

Las BSAs se unen mediante un sistema de distribución. El estándar IEEE 802.11 no especifica la tecnología usada para el sistema de distribución, aunque suele ser Ethernet. En la figura 2.3, podemos observar un ejemplo del área cubierta por un conjunto de servicios extendidos, a partir de la unión de varias BSS.

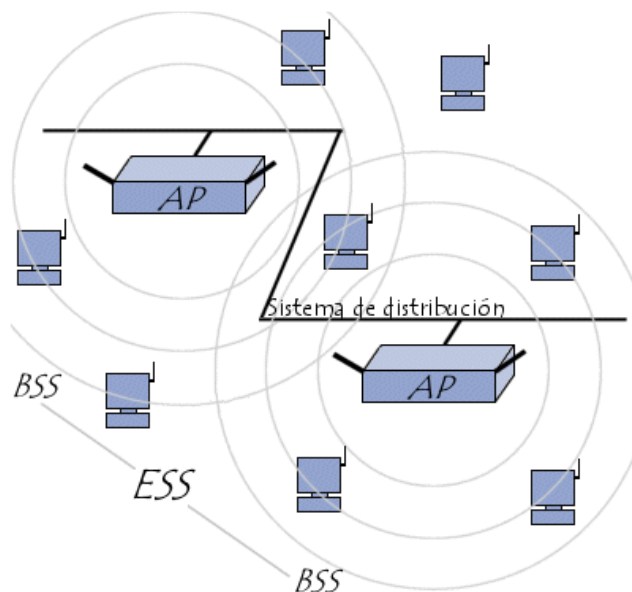


FIG2.3: EJEMPLO DE ESS

2.3 Arquitectura de capas

Las redes WLAN utilizan la arquitectura TCP/IP, formada por un conjunto de protocolos utilizados por los ordenadores conectados a Internet (o a otras redes, e.g. LAN). En Internet existen ordenadores y dispositivos de clases muy diferentes, con hardware y software incompatibles en muchos casos. Además, existen distintas formas de conectarse a Internet (Ethernet, WiFi, ...). TCP/IP no depende del equipo ni de la tecnología usada para conectarse a Internet. En este aspecto radica la gran ventaja de utilizar TCP/IP.

TCP/IP es una arquitectura de protocolos. Los dos protocolos más importantes son el Protocolo de Control de la Transmisión (TCP) y el Protocolo de Internet (IP), que son los que dan nombre al conjunto. La arquitectura de TCP/IP consta de cinco capas como se puede observar en la figura 2.4.

Capa de aplicación (VoIP, VoD,...)
Capa de transporte (TCP, UDP)
Capa de red (IP)
Capa de enlace
Capa física

FIG2.4: ARQUITECTURA TCP/IP

Estas cinco capas son, capa de aplicación, capa de transporte, capa de red, capa de enlace y capa física. Esta pila de protocolos es usada tanto por redes LAN cableadas como en redes WLAN. La diferencia que existe entre las distintas redes de área local estriba en la capa de enlace y en la tecnología usada en la capa física.

La capa de aplicación ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan para intercambiar datos. Ejemplos de protocolos de la capa de aplicación son VoIP (Voz sobre IP) y VoD (Video Bajo Demanda).

La capa de transporte, como se ha comentado anteriormente, se basa en los protocolos TCP o UDP (Protocolo de Datagramas de Usuario). Esta capa se encarga de la transferencia de datos libre de errores entre el emisor y el receptor, así como de mantener el flujo de red.

La capa de red utiliza el protocolo IP. La función de la capa de red es la del enrutamiento del tráfico y de asignar direccionamiento lógico a los dispositivos conectados a Internet.

La capa de enlace es la encargada de proporcionar direccionamiento físico a los dispositivos, acceso al medio, notificación de errores y control de flujo. En redes IEEE 802.x, la capa de enlace se subdivide en dos subcapas, la subcapa de Control de Enlace Lógico (LLC) y la subcapa de Control de Acceso al Medio (MAC). Todas las redes de área local usan el estándar IEEE 802.2 para la subcapa LLC. Sin embargo, el protocolo de la subcapa MAC y la capa física pueden variar ya que dependerán del tipo de LAN requerida, *Ethernet*, *Token Ring*,...

2.3.1 Capa física

El estándar IEEE 802.11 define la capa física (figura 2.5). Esta capa se subdivide en dos:

- Subcapa Dependiente del Medio Físico (PMD): esta subcapa se encarga de la modulación y codificación. Es dependiente de la tecnología concreta utilizada (infrarrojo o radio).
- Subcapa de Procedimiento de Convergencia (PLCP): trata las tramas generadas en la subcapa MAC y las adapta al formato adecuado para la subcapa PMD.

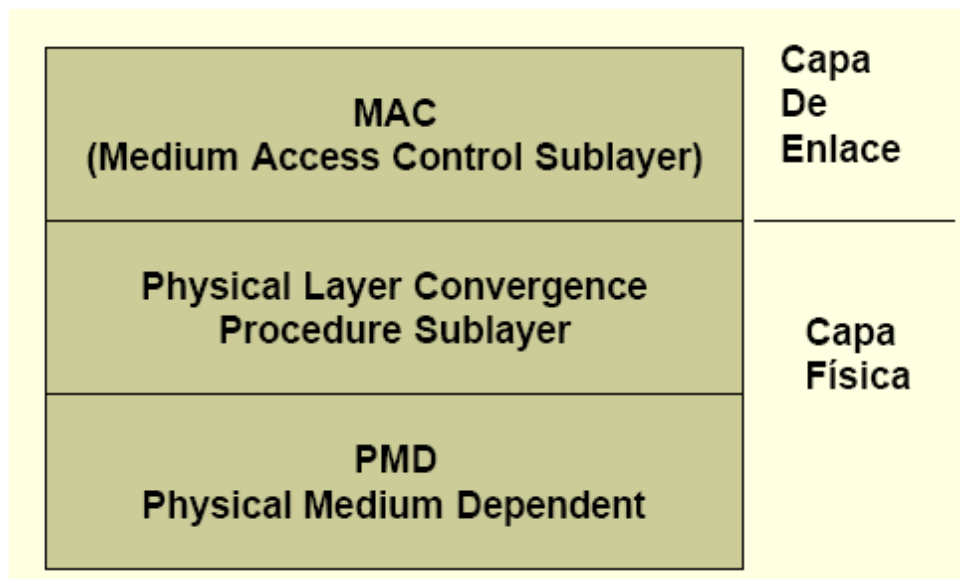


FIG2.5: CAPA FÍSICA DEL ESTÁNDAR IEEE 802.11

Estándares relacionados con la capa física

El interés por las redes inalámbricas ha posibilitado una rápida evolución del estándar inicial y actualmente existen tres extensiones del estándar original IEEE 802.11:

- IEEE 802.11a
- IEEE 802.11b
- IEEE 802.11g

Todas ellas trabajan en bandas de frecuencias que no necesitan licencia para su utilización: ISM (Industrial, Científico y Médicas) en 2.4GHz y U-NII (Infraestructura de Información Nacional sin Licencia) en 5GHz.

IEEE 802.11b es el estándar más usado, aunque su velocidad de transmisión máxima (11Mbps) es inferior a la alcanzada con el estándar IEEE 802.11a (54Mbps). Esto se debe a que el estándar IEEE 802.11a usa modulación OFDM (Multiplexación por División de Frecuencias Ortogonales) que utiliza un mayor número de bits por símbolo, y por tanto, obtiene una mayor tasa de transferencia efectiva. Sin embargo, debido a

que IEEE 802.11b utiliza una banda de frecuencia menor (2,4GHz), el alcance es aproximadamente el doble que con IEEE 802.11a, que trabaja en la banda de 5GHz. Posteriormente se incorporó un estándar que permitía transmitir datos a 54Mbps y que era compatible con el estándar IEEE 802.11b que sería denominado IEEE 802.11g.

IEEE 802.11

La primera versión del estándar IEEE 802.11 especifica dos velocidades de transmisión teóricas de 1Mbps y 2Mbps transmitidas por comunicaciones infrarrojas o por ondas de radio. El estándar original utiliza el protocolo CSMA/CA (Acceso Múltiple con Detección de Portadora Evitando Colisiones) como método de acceso. También se especifican las técnicas de modulación FHSS (Espectro Ensanchado por Salto de Frecuencia) y DSSS (Espectro Ensanchado por Secuencia Directa).

IEEE 802.11a

IEEE 802.11a opera en la banda de 5 GHz. Utiliza cincuenta y dos subportadoras OFDM con una velocidad máxima teórica de 54 Mbps (velocidades reales de aproximadamente 20 Mbps). La velocidad de transmisión se puede reducir considerablemente en función de las condiciones del canal. Este estándar utiliza doce canales no solapados, de los cuáles, ocho son para redes de infraestructura y cuatro para conexiones punto a punto.

IEEE 802.11a es incompatible con equipos que cumplan los estándares IEEE 802.11b ó IEEE 802.11g, ya que trabaja a distinta frecuencia.

Dado que la banda de 2.4 GHz se utiliza en múltiples tecnologías (teléfonos inalámbricos, bluetooth,...), el uso de la banda de 5 GHz representa una ventaja del estándar 802.11a ya que presenta menos interferencias. Sin embargo, los equipos que trabajan con este estándar tienen un radio de cobertura inferior a los que usan la banda de 2.4GHz

IEEE 802.11b

IEEE 802.11b funciona en la banda de 2.4GHz y tiene una velocidad máxima de transmisión de 11 Mbps. Debido al tiempo empleado por el protocolo CSMA/CA en el acceso al canal, en la práctica la velocidad máxima de transmisión es de aproximadamente 5.9 Mbps sobre TCP y 7.1 Mbps sobre UDP (véase [9]).

Aunque también utiliza una técnica de ensanchado de espectro basada en DSSS, introduce el esquema de modulación CCK (Modulación del código complementario) para llegar a velocidades teóricas de 5,5 y 11 Mbps. El estándar también incluye el uso de PBCC (Codificación Convolutiva del Paquete Binario) como esquema de modulación opcional.

Los dispositivos 802.11b deben mantener la compatibilidad con el equipamiento DSSS que cumpla la norma IEEE 802.11.

IEEE 802.11g

IEEE 802.11g se introduce como la evolución del estándar 802.11b. Utiliza la banda de 2.4 GHz, pero, gracias a la utilización de OFDM, opera a una velocidad teórica máxima de 54 Mbps (22 Mbps de velocidad real según [9]), igual que el estándar 802.11a. También puede usar la modulación DSSS.

Es compatible con el estándar IEEE 802.11b. Sin embargo, en este tipo de redes, la presencia de nodos bajo el estándar 802.11b reduce significativamente la velocidad de transmisión.

Los equipos que trabajan bajo el estándar 802.11g se incorporaron al mercado muy rápidamente. Esto se debió, en parte, a que para construirlos se podían adaptar los ya diseñados para el estándar 802.11b.

Resumen de los estándares IEEE 802.11

A continuación se expone en la tabla 2.1 un resumen con los datos más importantes de cada uno de los estándares IEEE 802.11:

Estándar	Banda	Tasa	Modulación
IEEE 802.11	2.4GHz	2Mbps	IR,FHSS, DSS
IEEE 802.11 a	5GHz	54Mbps	OFDM
IEEE 802.11 b	2.4GHz	11Mbps	DSS
IEEE 802.11 g	2.4GHz	54Mbps	DSS, OFDM

TABLA 2.1: RESUMEN DE LOS ESTÁNDARES IEEE 802.11

2.3.2 Capa de enlace

En IEEE 802.11, como se puede observar en la figura 2.6, la capa de enlace se subdivide en dos subcapas:

- Subcapa de Control de enlace lógico (LLC): esta capa se encarga del control de errores y del control del flujo. Además proporciona un interfaz tanto a la subcapa inferior MAC como a la capa superior (capa de red). En redes IEEE 802.x se usa el estándar IEEE 802.2 como protocolo de esta subcapa.
- Subcapa de Control de Acceso al Medio (MAC): entre las principales funciones se encuentran el control del acceso de los dispositivos al medio físico, el direccionamiento y la delimitación de tramas.

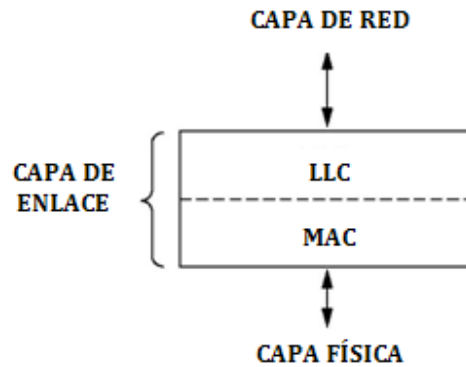


FIG2.6: CAPA DE ENLACE IEEE 802.11

El protocolo de acceso al medio usado en el estándar se denomina CSMA/CA. En este protocolo, las estaciones comprueban el medio antes de transmitir. A diferencia de CSMA/CD (Acceso Múltiple por Detección de Portadora con Detección de Colisiones), la estación emisora tiene que esperar una confirmación del receptor.

A continuación, se va a realizar una descripción de la capa MAC de IEEE 802.11, donde se encuentran las funciones PCF y DCF, que sirven como base de las funciones HCCA y EDCA del estándar IEEE 802.11e.

2.3.3 Capa MAC IEEE 802.11

La subcapa MAC, definida en IEEE 802.11, no tiene el modelo de servicios diferenciados. Este modelo mejora la gestión del tráfico generado por las nuevas aplicaciones multimedia. Además, en el estándar IEEE 802.11, se pierde tasa de bit en la fragmentación de paquetes, tiempo entre tramas y mensajes de confirmación. A lo anterior, se podría añadir que en condiciones de carga alta, las colisiones y tiempos de espera (*backoff*) pueden deteriorar gravemente la calidad de las aplicaciones de voz y video.

Por tanto, es necesaria la creación de nuevos estándares (IEEE 802.11e), con el objetivo de mejorar el rendimiento de las aplicaciones multimedia en redes WLAN. Para poder estudiar las mejoras de estos nuevos estándares, en primer lugar, se describe la arquitectura de la capa MAC del estándar original IEEE 802.11. Una vez expuesta ésta, se describirá el nuevo estándar IEEE 802.11e.

El protocolo MAC del estándar IEEE 802.11 tiene dos funciones de coordinación:

- La Función de Coordinación Distribuida (DCF)
- La Función de Coordinación Puntual (PCF).

Estas funciones determinan cuándo una estación puede transmitir ó recibir datos a través del medio inalámbrico.

DCF permite la transmisión de datos asíncronos de acuerdo con el método de mejor esfuerzo. Con DCF, todas las estaciones compiten por una oportunidad para poder transmitir.

Por otro lado, PCF permite la transmisión de datos síncronos. Usa un Coordinador Puntual (PC) para decidir que estación puede transmitir. Con PCF, se alternan períodos de contención (CP), durante los que las estaciones compiten por el canal, y períodos libres de contención (CFP), en los que el PC controla quién transmitirá (eliminando la necesidad de que las estaciones compitan por conseguir el acceso al canal).

2.3.3.1 Función de Coordinación Distribuida

En la actualidad, la mayoría de las redes WLAN utilizan DCF para acceder al medio. DCF está basado en un servicio de mejor esfuerzo y por tanto, no proporciona QoS. En este modo, la decisión de quien ocupa el medio se realiza de forma distribuida entre todas las estaciones. Las aplicaciones que usan DCF no especifican la prioridad al acceder al medio y, por tanto, no se garantiza QoS.

La función de coordinación distribuida permite compartir el medio inalámbrico entre las estaciones (en DCF no es necesario que haya un AP) usando el protocolo CSMA/CA. Éste soporta dos modos de funcionamiento.

- Modo básico
- Modo RTS/CTS.

Modo básico

1. Cada estación comprueba si el medio está libre antes de empezar a transmitir. Si el medio está libre, ha de esperar un intervalo de tiempo igual al Espacio Entre Tramas para DCF (DIFS, véase el apartado 2.3.3.3), tras el cual la estación comienza su transmisión.
2. Si una estación quiere transmitir estando el medio ocupado, la estación tendrá que esperar hasta que el medio esté libre durante un período DIFS. A continuación, comienza el mecanismo de *backoff*, que se usa para prevenir que dos o más estaciones transmitan simultáneamente. Para ello, la estación calculará un intervalo de tiempo aleatorio denominado intervalo de *backoff*, que tiene un valor entre cero y su Ventana de Contención (CW). El valor de CW depende del número de intentos en retransmisiones previas. CW tiene el mismo valor inicial para todas las estaciones IEEE 802.11 WLAN, por lo que todas las STAs tienen la misma probabilidad inicial de acceder al medio. Si tras el intervalo de *backoff* el medio sigue libre, la estación podrá comenzar a transmitir.
Con el mecanismo de *backoff*, se minimiza la probabilidad de colisión entre estaciones que quieran transmitir, ya que han de esperar un tiempo aleatorio (*backoff*) diferente tras el período DIFS (común para todas las estaciones).
3. DCF incluye mecanismos para la gestión de tramas erróneas o perdidas. Las estaciones receptoras deben transmitir una trama de confirmación (ACK) si el campo de comprobación (CRC) de la trama recibida es correcto. La estación

emisora esperará una trama ACK para confirmar la recepción correcta de la trama.

Si no se recibe el mensaje ACK, se considera como una indicación de trama perdida. Nótese, sin embargo, que si no se recibe una trama ACK puede deberse también a la pérdida de ésta y no a la de la trama original. Para comprobar esto, la estación emisora mantiene, para cada trama de datos, un temporizador para considerar la trama como pérdida si no ha llegado su confirmación. Por otro lado, hay que indicar que para enviar una trama ACK, hay que esperar antes un Espacio Entre Tramas Corto (SIFS, véase el apartado 2.3.3.3), que es menor que DIFS, y que tiene mayor prioridad para transmitirse que una trama de datos.

4. Cuando se detecta que una trama originada desde otra estación no ha sido recibida correctamente (no llega su confirmación), las retransmisiones han de utilizar un Espacio Entre Tramas Extendido (EIFS, véase el apartado 2.3.3.3), en lugar de DIFS. El intervalo EIFS tiene una duración superior al período DIFS (las retransmisiones tienen menor prioridad que las transmisiones iniciales). Además, IEEE 802.11 contempla un parámetro configurable que define el número máximo de retransmisiones hasta recibir una trama ACK.

La figura 2.7, muestra un ejemplo de funcionamiento del modo básico de DCF:

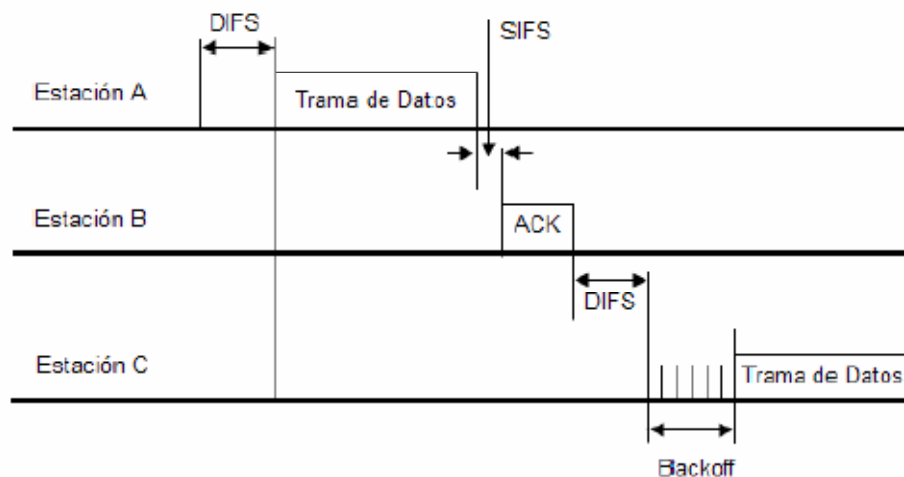


FIG2.7: FUNCIONAMIENTO MODO BÁSICO

En este ejemplo, la estación A tiene que esperar un DIFS antes de poder transmitir una trama. Una vez que la estación A ha terminado de emitir su trama, espera una trama ACK de la estación B que permite confirmar la recepción. La trama ACK se envía tras un intervalo de tiempo denominado SIFS, inferior a DIFS. Si el temporizador asociado expira y en ese tiempo no hubiese llegado la confirmación, la estación emisora debe considerar que ha habido un error en la transmisión o que se ha producido una colisión.

Supóngase que en algún momento durante la emisión de la trama de A, una estación C desea enviar una nueva trama. Como C detecta que el canal está ocupado, esperará hasta que el canal este de nuevo libre. Una vez que B ha enviado la confirmación a la estación A, el canal volverá a estar disponible para que otra estación lo ocupe. Entonces C lo detectará y esperará un tiempo DIFS para comprobar si sigue vacío. Si es así, C no transmitirá de inmediato, sino que ejecutará el algoritmo de *backoff* esperando un tiempo aleatorio. Si durante el tiempo aleatorio C detecta que alguna estación transmite, congelará su contador de tiempo aleatorio para volver a activarlo tras un tiempo DIFS después de que haya cesado toda actividad. La transmisión de la trama se hará cuando el contador llegue a cero.

Modo RTS/CTS

Se basa en el protocolo MACAW (*Multiple Access with Collision Avoidance for Wireless*) y usa detección del canal virtual.

Este método se utiliza para evitar el problema del nodo oculto. Este problema ocurre cuando una estación quiere transmitir y no puede saber el estado de otra estación lejana. En ese caso cree que el medio está libre y provoca una colisión (llegada simultánea de dos paquetes al receptor).

La figura 2.8 presenta un ejemplo del funcionamiento de este método.

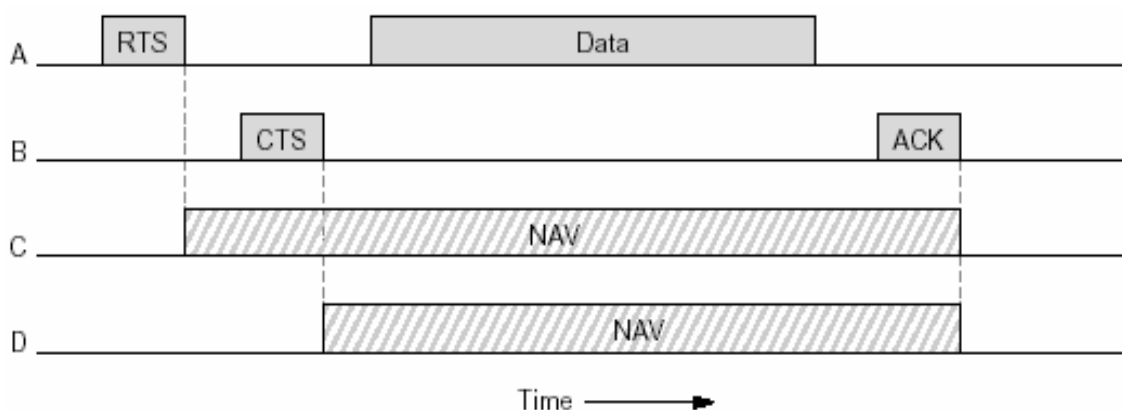


FIG2.8: FUNCIONAMIENTO MODO RTS/CTS

La estación A quiere transmitir a la B. La estación C, está en radio de alcance de la A. Por otro lado, la estación D está en el radio de alcance de la B. Véase la figura 2.9.

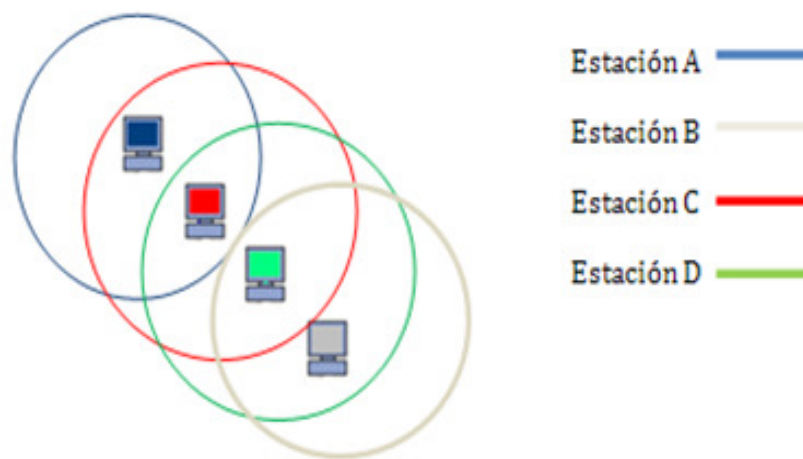


FIG2.9: TOPOLOGÍA RED INALÁMBRICA

1. A envía un paquete *Request To Send* (RTS) a B, que contiene el tamaño de la trama de datos que quiere transmitir.
2. Si decide concederle permiso a A para transmitir, B responde con un paquete *Clear To Send* (CTS) que contiene una copia del tamaño de la trama de datos que A quiere transmitir.
3. A envía su trama e inicia su temporizador de espera del paquete de ACK, que le indicará que su trama fue recibida satisfactoriamente.
4. Al recibir la trama de datos correctamente, B envía a A el mensaje de ACK.

Considérese lo que sucede desde el punto de vista de C y D:

1. Como C está en el radio de alcance de A, recibe el paquete *Request To Send*, percibiendo que alguna estación quiere iniciar una transmisión. A partir de la información contenida en el paquete *Request To Send* estima cuánto tardará la transmisión de la trama de datos y su confirmación, por lo que se impone para sí misma un tipo de canal virtual ocupado, indicado por el Vector de Asignación de Red (NAV en la figura 2.8). NAV no se transmite, sólo sirve de recordatorio para mantenerse en silencio durante ese período y poder así llevar a cabo la transmisión de datos entre A y D sin colisiones.
2. Como D está en el radio de alcance de B escucha el paquete *Clear To Send*, por lo que también se impone un NAV para sí misma.

2.3.3.2 Función de coordinación puntual

La función de coordinación puntual es otro método de acceso al canal del estándar IEEE 802.11. Proporciona calidad de servicio, ya que fue diseñada para soportar servicios con restricciones temporales y para permitir que las estaciones tengan acceso

con diferentes prioridades al medio inalámbrico. La función PCF la realiza el coordinador puntual, normalmente ubicado en el punto de acceso.

PCF realiza el acceso al canal siguiendo los siguientes pasos (véase la figura 2.10):

- Para que una estación inicie una comunicación en una red IEEE 802.11, usando la función PCF, primero tiene que asociarse a la red. La asociación es un proceso en el cual el punto de acceso reserva recursos (memoria, tasa de transmisión,...) para una estación. Además, se envía información importante como, por ejemplo, las tasas de transmisión soportadas y el Identificador de Conjunto de Servicios (SSID) o nombre de la red que debe ser el mismo en todos los elementos conectados a la misma red. Finalmente, el punto de acceso responde a la estación con una trama de respuesta de asociación.
- Una vez hecha la asociación, el AP añade la estación a una lista de turnos con las estaciones asociadas. Una STA puede transmitir cuando recibe una trama de asignación de turno (trama CF-Poll o de *polling*) del AP. Cada CF-Poll permite transmitir una única trama.
- El comienzo de un período libre de contención viene definido por la transmisión, por parte del AP, de una trama *beacon*. Estas tramas son generadas en intervalos de tiempo periódicos y conocidos por el resto de estaciones. Uno de los datos aportados por esta trama es la duración máxima del período libre de contención, *CFPMaxDuration*.
- Tras la recepción de la trama *beacon*, las estaciones tendrán que esperar a que el AP les envíe una trama CF-Poll antes de poder transmitir. Cada transmisión en el intervalo libre de contención está separado por un SIFS, para evitar colisiones entre transmisiones.
- Además, ninguna estación que use DCF puede conseguir el medio durante este período ya que el intervalo PIFS es menor que el espacio entre tramas DIFS, como ya se comentó anteriormente.
- En el caso de que una estación no necesitara transmitir ninguna trama, igualmente, el AP le envía un mensaje de *polling*. Esta estación mandaría una trama sin datos indicando al AP su situación. Después de esperar un intervalo PIFS, el PC le enviaría una trama de *polling* a la siguiente estación de su lista. De esta forma, el PC conserva la prioridad para acceder al medio ya que el intervalo PIFS es inferior al intervalo SIFS. Si el PC no recibe una respuesta de la segunda estación, espera otro intervalo PIFS y envía una trama de *polling* a la tercera estación de la lista y así sucesivamente hasta encontrar una estación que quiera transmitir.
- Por último, además de las tramas de *polling* y de ACK, el PC también envía tramas de gestión (potencia, sincronización, asociación, etcétera).

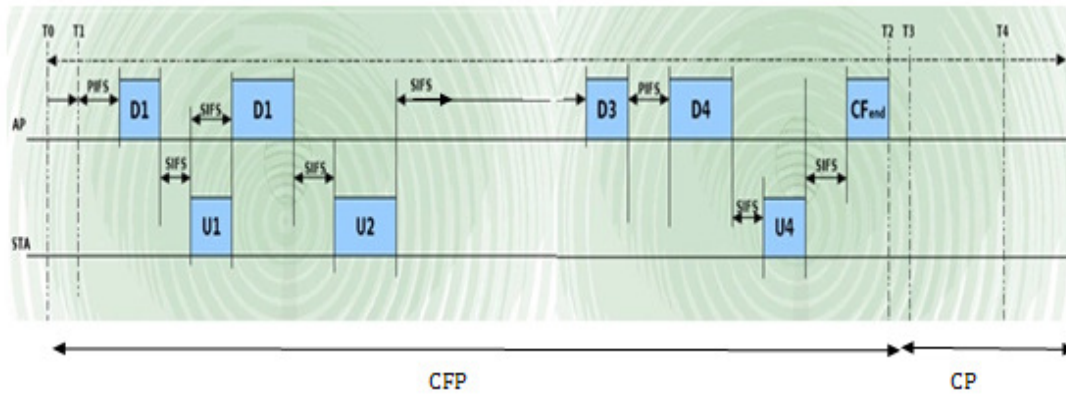


FIG2.10: EJEMPLO TRÁFICO IEEE 802.11

Como resumen se muestra en la figura 2.11, un resumen con los principales puntos comentados con anterioridad:

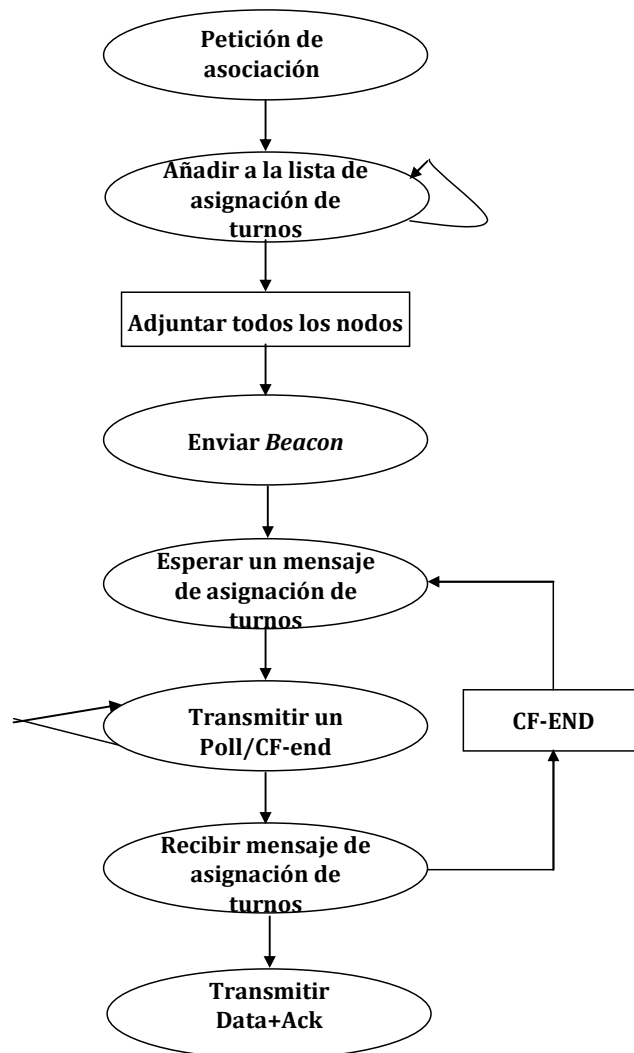


FIG2.11: ALGORITMO PCF

A continuación se van a comparar algunos aspectos de los dos métodos de acceso al canal del estándar IEEE 802.11.

En primer lugar, la función de coordinación distribuida es obligatoria en el estándar IEEE 802.11 y como consecuencia de esto, es el modo más usado. Por otro lado, la función de coordinación puntual es opcional, pero puede coexistir con el modo distribuido y permitir conexiones de estaciones que usen únicamente DCF.

La segunda diferencia se encuentra en el espacio entre tramas de cada uno de los métodos de acceso al canal. En el modo PCF las estaciones esperan un tiempo entre tramas, llamado espaciado entre tramas PCF (PIFS, véase el apartado 2.3.4.1). Éste es inferior al esperado en el modo DCF (DIFS), y por consiguiente, la función PCF tiene prioridad respecto a la función distribuida para acceder al medio.

Por último, hay que comentar los mecanismos que tiene cada uno de los métodos para evitar colisiones en el medio inalámbrico. En el caso de la función de coordinación puntual, el PC asigna turnos a las distintas estaciones mediante el envío de un mensaje de asignación de turnos (*polling*) a cada una de ellas, indicándoles que pueden transmitir sus tramas. Usando este método, PCF evita que se produzcan colisiones. Por su parte, DCF utiliza un tiempo fijo DIFS y un tiempo aleatorio de *backoff* para evitar las colisiones cuando las estaciones compiten por el canal (período de contienda o contención). Debido a esta aleatoriedad, en DCF si es posible que haya colisiones entre las transmisiones de estaciones diferentes.

Tramas PCF

En IEEE 802.11, para mejorar la eficiencia, se pueden combinar tramas de información de control, de confirmación, de asignación de turnos y de transferencia de datos uniendo varias funciones en una sola trama. De esta forma, se puede enviar en una misma trama una respuesta de recepción de las tramas anteriores, transmitir el mensaje de *polling* a otra estación diferente, etc.

Dado que PCF es la función en la que se basa HCCA (función utilizada por todos los algoritmos estudiados en el proyecto), se presentan las tramas de este mecanismo (en el período libre de contención):

Datos

El PC envía esta trama a una estación cuando tiene que enviarle información y confirmación a una transmisión anterior. No incluye asignación de turnos, por lo que el receptor no puede transmitir una trama. La trama de datos se utiliza en el período libre de contención y es idéntica a la trama de datos usada en los períodos con contención.

CF-Ack

Esta trama se envía para confirmar que una trama de datos ha llegado correctamente y sin errores, siempre que el receptor no necesite transmitir algún dato. En ese caso se usa la trama Datos+CF-Ack .

CF-Poll

El PC envía un CF-Poll a una estación para darle la posibilidad de transmitir una trama. Se utiliza cuando el punto de acceso no tiene ningún dato para la estación. Cuando éste tiene alguna trama disponible para la estación, el punto de acceso utiliza una trama Datos+CF-Poll. Además, esta trama se utiliza si el PC tampoco necesita reconocer la recepción de tramas anteriores, en cuyo caso utilizaría *CF-ACK+CF-Poll*.

Un escenario práctico es el que se observa en la figura 2.12. El punto de acceso transmite un CF-Poll y la estación sondeada no tiene ningún dato. Esta responde enviando al PC una trama nula (esta trama no se muestra en la figura). Posteriormente, el PC coge otra estación de la lista de *polling*. Si no dispone de datos para transmitir a la segunda estación, le transmite un mensaje CF-poll.

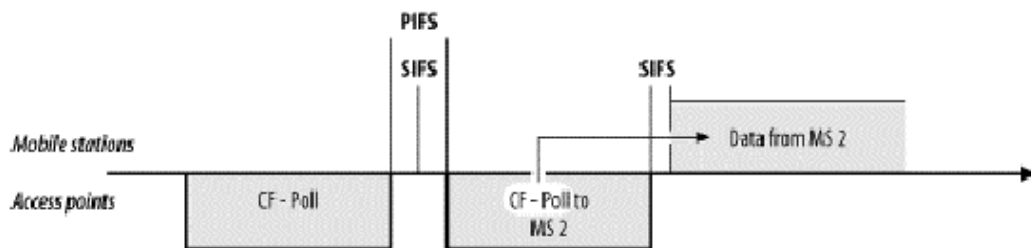


FIG2.12: TRAMA CF-POLL

Datos+CF-Ack

Esta trama combina las transmisiones de datos con un mensaje de confirmación. Los datos son enviados al destinatario de la trama; el mensaje ACK es el de respuesta a la trama transmitida anteriormente y generalmente no es para el receptor de los datos.

En la figura 2.13, se puede observar que el AP le envía un mensaje de *polling* a la estación 1, y ésta tras un SIFS, envía los datos y confirma la trama anterior.

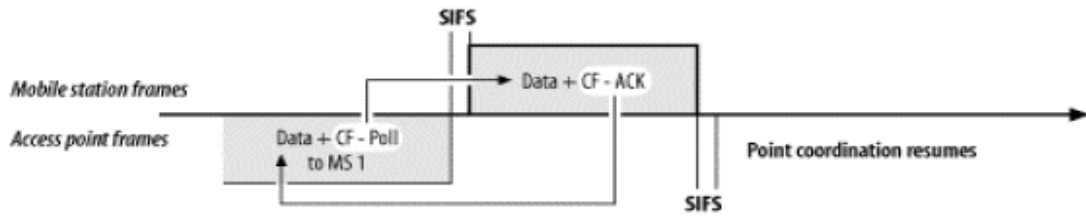


FIG2.13: TRAMA DATOS+CF-ACK

Datos+CF-Poll

Esta trama la utilizan los PC para transmitir datos a una estación móvil y solicitar una trama a otra estación. La trama de Datos+CF-Poll se puede enviar solamente durante el período libre de contención y sólo cuando el AP no necesita enviar un mensaje ACK para otra trama.

CF-ACK+CF-Poll

Esta trama es un mensaje de confirmación de la última trama enviada al PC y solicita además una trama del siguiente nodo en la lista de *polling*. Esta trama se dirige a la próxima estación de la lista, aunque la confirmación puede ir dirigida a cualquier estación móvil asociada con el PC (véase la figura 2.14).

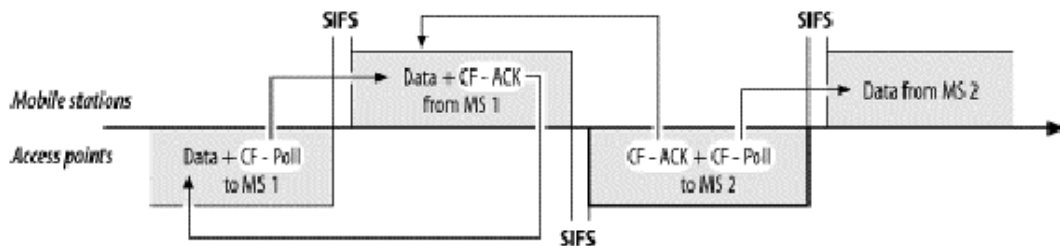


FIG2.14: TRAMA CF-ACK+CF-POLL

Datos+CF-ACK+CF-Poll .

Esta trama reúne la transmisión de datos, un mensaje de *polling*, y una confirmación en una trama para lograr la máxima eficiencia. La siguiente figura 2.15 ilustra el uso de Datos+CF-Ack+CF-Poll. Como sucede también con la trama Datos+CF-Ack, los componentes de esta trama son descifrados generalmente por diversas estaciones. La transmisión de datos y el mensaje de *polling* se dirigen a la misma estación, pero el mensaje de ACK es para la estación de la transmisión anterior, que no tiene por qué coincidir con la dirección de los datos y el mensaje de *polling*.

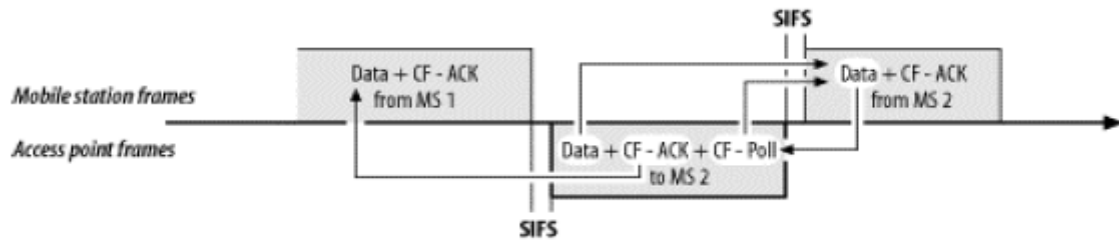


FIG2.15: TRAMA DATOS+CF-ACK+CF-POLL

La figura 2.15 comienza con la estación móvil 1 (MS1) transmitiendo una trama Datos+CF-Ack. Los datos van dirigidos al punto de acceso, y se utiliza una trama CF-Ack para confirmar la trama de datos transmitida anteriormente por el punto de acceso (esta trama no se muestra en la figura). El punto de acceso envía entonces un mensaje de *polling* a la estación móvil 2 (MS2). Además el punto de acceso debe confirmar los datos de MS1, y por eso transmite una trama con un CF-Ack. Cuando el AP también tiene datos para transmitir, las tres características pueden combinarse en una trama y se envía una trama Datos+CF-Ack+CF-Poll. El destino de la trama de datos recibe los datos y el mensaje CF-Poll, pero el CF-Ack es para el transmisor de la trama anterior. MS1 debe escuchar las tramas del punto de acceso para obtener una confirmación de los datos.

CF-Fin

Esta trama se envía para finalizar el período libre de contención y devuelve el control del medio a la función de coordinación distribuida.

CF-Fin+CF-Ack

Esta trama tiene la misma función que el mensaje CF-Fin, pero además incluye un mensaje ACK de la trama de datos transmitida anteriormente.

Cualquier trama de control

No se pone ninguna restricción en el estándar a las tramas de control que pueden transmitirse durante el período libre de contención. Si las reglas que se aplican a un tipo particular de trama permiten su transmisión, el punto de acceso puede transmitirlo.

Trama nula

Una estación transmite una trama nula en respuesta a un mensaje de *polling* del AP cuando no tiene datos para enviar.

Supertrama IEEE 802.11

Otro aspecto importante, una vez conocidos los dos métodos de acceso, es el de cómo se forma una supertrama IEEE 802.11 (véase la figura 2.16). El tiempo de transmisión se divide en supertramas que se componen de intervalos libres de contención (CFP) e intervalos con contención (CP). En el intervalo CFP se usa la función de coordinación puntual, y en el intervalo CP la función de coordinación distribuida.

El comienzo de una supertrama lo indica una trama llamada *beacon*. El punto de acceso envía esta trama periódicamente para difundir su presencia y enviar información de la red a las estaciones en su radio de cobertura. De una trama *beacon* se puede obtener información sobre cuáles son los AP disponibles, las características de la red, información de cómo poder conectar con el punto de acceso deseado y la duración de los períodos CFP y CP. Las tramas *beacon* también se usan en el caso de utilizar únicamente DCF.

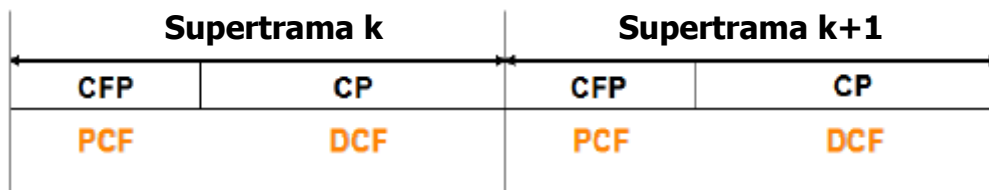


FIG2.16: SUPERTRAMA IEEE 802.11E

PCF se asemeja a los protocolos de redes basados en permiso o (*token*). En este caso, el PC toma la decisión de a qué estación le envía el *polling*, equivalente al *token* o testigo en protocolos o redes basados en permiso.

El tamaño relativo del período libre de contención es configurable. Por otro lado, la longitud mínima del período de contención es el tiempo requerido para transmitir al menos una trama con el tamaño máximo posible y recibir su ACK correspondiente de todas las estaciones asociadas. Sin embargo, es posible para el servicio basado en contención sobrepasar el final del período de contención. Cuando esto ocurre, y el CP sobrepasa el punto esperado de comienzo del período libre de contención, se dice que éste está ensombrecido (*foreshortened*) y se permite la transmisión completa de la trama iniciada durante el intervalo CP, antes de que la trama *beacon* anuncie el comienzo del período libre de contención. Esto provoca que el período libre de contención se acorte debido a los retrasos. El final del CFP termina como máximo en el límite anunciado por la trama *beacon*. Este límite se llama *Target Beacon Transmisión Time* (TBTT, tiempo entre transmisión de dos tramas *beacon*).

El PC también puede terminar el período libre de contención antes de su TBTT, transmitiendo una trama de CF-Fin. Puede basar esta decisión en el tamaño de la lista de *polling*, o en cualquier otro factor que el punto de acceso considere importante.

2.3.3.3 Espaciado entre tramas

En IEEE 802.11 para poder transmitir una trama, una vez que el medio se ha quedado libre, hay que esperar un período de tiempo. Hay cuatro tipos de tiempo o espacio entre tramas en el estándar IEEE 802.11 para el acceso al canal:

- Espacio entre tramas corto (SIFS): tiempo de espera antes de realizar transmisiones de alta prioridad (e.g. tramas de confirmación). También se usa para transmisiones de las estaciones durante el período libre de contención. Al ser el menor de los intervalos esperados, no son susceptibles de tener colisiones con otras transmisiones.
- Espacio entre tramas PCF (PIFS): tiempo de espera antes de poder transmitir una trama durante el período libre contención. Como el intervalo de espera en PCF es inferior al de DCF, estas transmisiones tendrán prioridad durante dicho período.
 - $PIFS = SIFS + SlotTime$
- Espacio entre tramas DCF (DIFS): tiempo mínimo que el medio debe estar libre antes de poder transmitir en el período de contención.
 - $DIFS = SIFS + 2 * SlotTime$
- EIFS (Espacio entre Tramas Extendido): espacio usado para las retransmisiones de tramas. No tiene un valor fijo, pero su valor es superior al resto de espacios entre tramas.

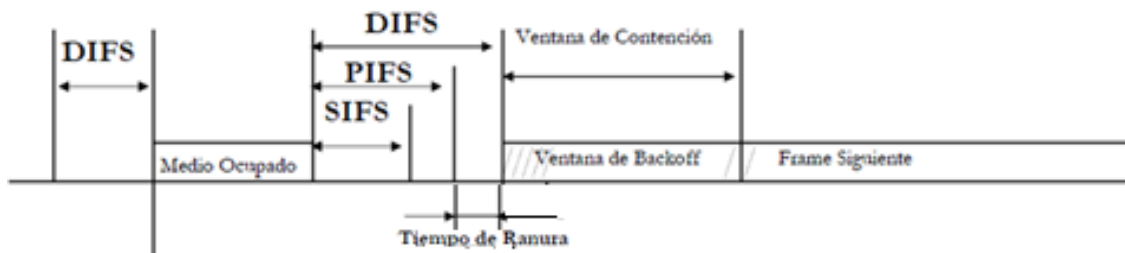


FIG2.17: ESPACIO ENTRE TRAMAS IEEE 802.11

A continuación se expone la tabla 2.2 con los valores de estos intervalos para los estándares IEEE 802.11 más habituales, así como el tamaño de sus ventanas de congestión:

Estándar	SlotTime (μ s)	SIFS (μ s)	DIFS (μ s)	PIFS (μ s)	CW min	CW max
802.11a	9	16	34	25	15	1023
802.11b	20	10	50	30	31	1023
802.11g	9 o 20	10	28 o 50	19 o 30	15 o 31	1023

TABLA 2.2: VALORES DE LOS ESPACIOS ENTRE TRAMAS IEEE 802.11

Como se puede comprobar en la tabla 2.2, el estándar IEEE 802.11g tiene la posibilidad de usar los mismos valores para los IFS que el estándar IEEE 802.11b, debido a la compatibilidad entre ellos.

2.3.3.4 Conclusión sobre PCF y DCF

A pesar de las ventajas que tiene PCF sobre DCF, la principal de ellas es que no hay contención y por tanto parece adecuado para servicios con restricciones temporales. Sin embargo, existe una serie de inconvenientes que impiden que garantice unos requisitos mínimos para las aplicaciones y servicios. Ni DCF, que está basado en contención, ni PCF tienen suficiente funcionalidad para dar QoS a aplicaciones multimedia como *streaming* de audio o video. DCF gestiona todos los tipos de tráfico de la misma forma, con todas las estaciones compitiendo por el medio con la misma prioridad. PCF, aunque está libre de contención, tiene una serie de incompatibilidades para soportar QoS, las cuáles se listan a continuación:

- El mecanismo de *polling* sigue un algoritmo *round robin*, que es un simple retardo equitativo de turnos, no demasiado útil para aplicaciones con diferentes requisitos de QoS
- Los retardos provocados por el envío de tramas *Beacon* son impredecibles.
- No hay un tiempo de transmisión máximo de las estaciones que han recibido un mensaje de *polling*, por lo que el retardo máximo hasta la siguiente transmisión no está acotado.
- No hay tráfico diferenciado con distintas prioridades.
- No hay un mecanismo para que las estaciones pueden comunicar sus requerimientos de QoS al AP.

3. IEEE 802.11e

La capa MAC especificada en el estándar IEEE 802.11 no es eficaz para aplicaciones con requisitos de tiempo real, tales como VoIP o videoconferencia. El estándar IEEE 802.11 no hace distinción, a la hora de repartir los turnos de transmisión, entre el tráfico de las distintas aplicaciones, ya sean éstas aplicaciones con requisitos de retardo y *jitter* acotado, o aplicaciones Web, correo electrónico o acceso remoto. En definitiva, IEEE 802.11 se usa para aplicaciones sobre TCP, que no requiere QoS.

Debido al incremento del número de servicios multimedia, se hace necesario poder soportar aplicaciones con QoS y sin requisitos de calidad de servicio sobre las mismas redes. El estándar IEEE 802.11e se ha propuesto para superar esta limitación. Las mejoras de IEEE 802.11e se pueden resumir en los siguientes puntos:

- Reducir la latencia o retardo dando prioridad a unos paquetes frente a otros dependiendo del tráfico actual.
- Capacitar al punto de acceso para planificar la gestión de los recursos, basándose en parámetros tales como la tasa de bit y el retardo.
- Mejorar la eficiencia en el uso del ancho de banda y reducir el *overhead* (pérdida de ancho de banda por exceso de bits de cabecera) de los paquetes.

Hay que comentar también que IEEE 802.11e introduce una nomenclatura nueva para poder diferenciarlo de IEEE 802.11:

- QSTA (Estación con QoS): estación con soporte de calidad de servicio.
- QAP (Punto de Acceso con QoS): punto de acceso con soporte de calidad de servicio.
- QBSS (Conjunto de Servicio Básico con QoS): BSS con soporte de calidad de servicio.

Además, en este estándar se define la función de coordinación híbrida que integra dos mecanismos de acceso al canal:

- Acceso Coordinado y Distribuido Mejorado (EDCA): es una ampliación de DCF para proveer QoS. Está basado en contención por lo que se usa en el

período con contención de la supertrama. Además es distribuido ya que la decisión de quien recibe el turno de transmisión se toma entre todos los nodos. EDCA está pensado para dar un servicio diferenciado, no garantizando requisitos de QoS ya que sólo da prioridad a unos tráficos frente a otros.

- Acceso al Canal Controlado por HCF (HCCA): es una ampliación de PCF para proporcionar QoS. Se puede usar tanto en el intervalo de contención como en el intervalo libre de contención de la supertrama. El acceso al canal está basado en el envío de mensajes de *polling*, siendo una función centralizada. El Coordinador Híbrido (HC), función que normalmente es desempeñada por el QAP, es el que decide a que estación le toca transmitir. Para ello, el HC envía mensajes de *polling* a las estaciones que tiene asociadas. Es una función que garantiza los requisitos de QoS.

Por tanto, ambas funciones de acceso mejoran o extienden la funcionalidad de los métodos de acceso del estándar IEEE 802.11 (DCF y PCF).

La unidad básica de asignación de recursos en estas funciones de acceso al canal es la oportunidad de transmisión (TXOP). Una TXOP es un intervalo de tiempo limitado durante el cual una QSTA puede transmitir una serie de tramas. El período TXOP se define con un tiempo de inicio y una duración máxima. Si el período TXOP se obtiene usando el acceso al canal basado en contención, entonces se denomina EDCA-TXOP. Si, por el contrario, se obtiene a través de HCCA se llama HCCA-TXOP. La duración del período EDCA-TXOP se controla a través del QAP y se transmite al resto de QSTAs en las tramas beacon junto con otros parámetros relacionados con EDCA. La duración del período HCCA-TXOP se transmite a las estaciones QSTA directamente en las tramas de polling por el HC. Además, una TXOP termina cuando:

- La estación o el QAP no tiene más datos que transmitir.
- Expira el tiempo de la TXOP actual.
- Para HCCA, cuando la estación que está transmitiendo deja libre el canal durante más de un intervalo de tiempo DIFS.

En la mayoría de los esquemas se alternan intervalos EDCA con HCCA al igual que ocurría con IEEE 802.11, como se puede observar en la figura 3.1:

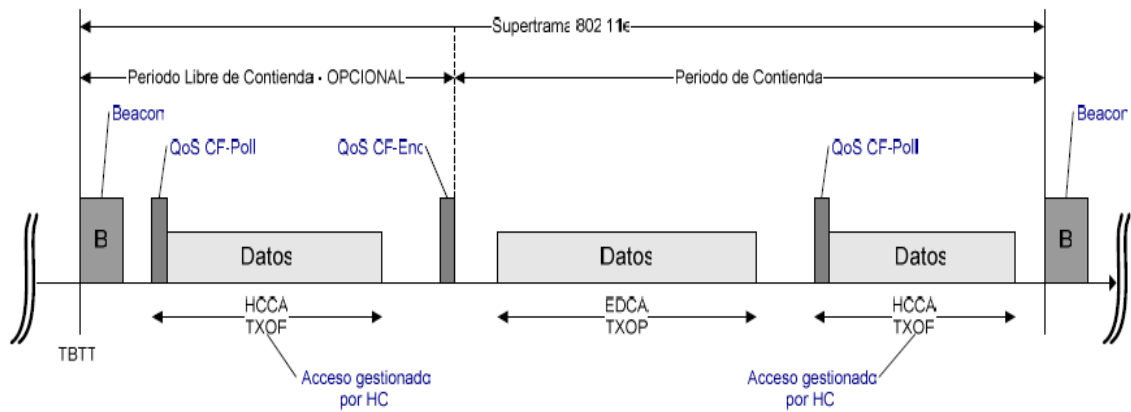


FIG3.1: ESQUEMA IEEE 802.11E

En éste proyecto se ha utilizado HCCA, es decir, se usará un sistema centralizado. Se ha elegido debido a que, como se comentaba en la introducción de este apartado, garantiza que se cumplen con los requisitos de QoS. EDCA sólo nos proporciona relativa diferenciación entre los distintos tipos de tráfico, ya que no nos garantiza un retardo y un throughput acotado. Sólo se utilizará EDCA para tratar con tráfico sin requerimiento de QoS (TCP).

Por otro lado, HCCA no especifica ningún algoritmo efectivo para el uso del ancho de banda, aunque sí que define un algoritmo de referencia básico. Este algoritmo sigue una planificación siguiendo un esquema round-robin. Por tanto, es necesario diseñar un nuevo algoritmo capaz de adaptarse a situaciones con cambios en la tasa de bit, en el retardo y en el jitter del tráfico.

3.1 Calidad de servicio

Calidad de Servicio (QoS) en una red se refiere a "una garantía de la red para satisfacer un desempeño predeterminado para un conjunto de servicios al usuario" basándose en parámetros cuantitativos tales como: el *Throughput*, el retardo extremo a extremo (latencia), variación en el retardo (jitter), pérdida de paquetes, etcétera

En el diseño de redes, la administración de recursos es extremadamente importante ofrecer a los usuarios finales y a sus aplicaciones, las condiciones adecuadas para la transmisión de sus flujos de tráfico. Debido a las nuevas aplicaciones, con requisitos de QoS, es necesario que existan mecanismos que garanticen QoS. Para cumplir con este objetivo, es imprescindible la correcta administración de los recursos en la red para que el tráfico pueda fluir de manera correcta. Por tanto, la provisión de calidad de servicio en la red es un objetivo de gran importancia.

Las aplicaciones multimedia tienen las siguientes características:

- Necesitan un ancho de banda mínimo.
- Son muy sensibles al retardo extremo a extremo.
- Son sensibles a la variación del retardo (*jitter*).

- Relativamente tolerantes a pérdidas de paquetes (<1% es tolerable).

Además, hay que integrar las aplicaciones multimedia con aplicaciones tradicionales que son elásticas (sin requisitos de QoS) y que no necesitan cumplir con los requisitos anteriores. Este es el objetivo principal del algoritmo propuesto WTTP, como se explicará posteriormente.

Por tanto, la calidad de servicio se puede definir como el proceso de entrega de datos de una forma fiable, incluyendo aspectos importantes como son la pérdida de paquetes y el retardo. Estos deben de ser mínimos o casi nulos, principalmente para aplicaciones en tiempo real, ya que los retardos y la pérdida de paquetes afecta en gran medida a estos servicios (e.g. VoIP, VoD).

Existen una serie de parámetros que definen la calidad de servicio y que se listan a continuación:

Paquetes perdidos

La pérdida de paquetes de datos se produce cuando éstos no llegan a su destino final. Puede ser debido a diferentes motivos. Por ejemplo, cuando se congestiona la red (los *routers* llenan de paquetes sus *buffers* y los desbordan) se pierden paquetes. Otro factor que puede provocar pérdida de paquetes son las interferencias en el medio inalámbrico, debido a la existencia de dispositivos que operan sobre la misma banda de frecuencias. El porcentaje de paquetes perdidos o desechados se puede obtener a partir de la expresión 1.

$$P_{PP} = \frac{P_P}{P_T} \times 100 \quad (1)$$

Donde:

P_{PP} = Porcentaje de paquetes perdidos.

P_P = Paquetes perdidos.

P_T = Paquetes transmitidos.

$$P_P = P_T - P_R \quad (2)$$

Donde:

P_P = Paquetes perdidos.

P_T = Paquetes transmitidos.

P_R = Paquetes recibidos.

Retardo

El retardo extremo a extremo o latencia, es el tiempo total que transcurre desde que un paquete de datos es generado en un nodo fuente hasta que es recibido por el nodo destino. Este parámetro se mide en unidades temporales.

En las aplicaciones de tiempo real como *streaming* de audio y video, es necesario que exista un retardo acotado para poder obtener una buena calidad de la aplicación.

Variación en el retardo o *jitter*

El *jitter* es la variación en el retardo que sufren los diferentes paquetes de un mismo flujo en el destino. Es debido a que, aunque los paquetes enviados pueden seguir una misma trayectoria dentro de la red, se van a ver afectados por factores distintos (almacenamiento de los datos en una cola, condiciones del medio inalámbrico, asignación de turnos, etcétera).

Tasa de transferencia efectiva (*Throughput*)

La tasa de transferencia efectiva es la velocidad con la que es recibido el tráfico en un nodo destino. Se puede obtener a partir de la expresión 3.

$$C_E = \frac{D_R}{T_T} \quad (3)$$

Donde:

C_E = *Throughput*.

D_R = Datos recibidos.

T_T = Tiempo total.

Para cumplir con los requisitos de retardo, *jitter* y la tasa de transferencia efectiva, el HC tiene un procedimiento de reserva de recursos para los flujos de tráfico que necesitan QoS. El HC decidirá mediante el control de admisión cuáles de estos flujos serán admitidos y cuáles no. Esto se hará en base al TSPEC (especificación del tráfico) que se envía al HC desde cada una de las QSTAs.

El TSPEC es un conjunto de parámetros que definen las características de un flujo de tráfico de una determinada QSTA. Es importante conocer los principales parámetros que proporciona el TSPEC:

- Tipo de tráfico: nos indica si el tráfico es periódico o no periódico.
- Identificación del Tipo de Flujo (TSID): la QSTA identifica cada uno de sus flujos por un número.
- Dirección: parámetro que indica si se trata del enlace ascendente o descendente.
- Tasa de Datos Promedio (ρ): tasa de bit promedio que se requiere para la transferencia de las tramas.
- Retardo (D): tiempo máximo para el transporte de tramas que incluye la propagación por el medio inalámbrico y el retardo en la cola.

- Tamaño Nominal de la SDU (N): tamaño promedio de la unidad de datos de servicio (SDU).
- Tamaño Máximo de la SDU (M): tamaño máximo de la SDU.
- Mínimo Intervalo de Servicio (mSI): mínimo intervalo de tiempo entre el comienzo de períodos de servicio sucesivos.
- Máximo Intervalo de Servicio (MSI): máximo intervalo de tiempo entre el comienzo de períodos de servicio sucesivos.
- Tamaño de la Ráfaga (BS): máximo tamaño de ráfaga de datos de un flujo determinado.
- Tasa Mínima en la Capa Física (R): mínima tasa de bit en el medio inalámbrico.
- Tiempo de Comienzo del Servicio (SST): indica el momento en el que el intervalo de servicio comienza.

El algoritmo de planificación que especifica el estándar IEEE 802.11e a modo de ejemplo (planificador de referencia) utiliza varios de los parámetros del TSPEC para repartir el ancho de banda. En concreto, usa "MSI" o "D", "p" y "L". En este algoritmo, el HC decide primero un intervalo de servicio común a todos los flujos de tráfico. El intervalo de servicio toma el valor del mayor submúltiplo del intervalo de *beacon* (supertrama). Además, este valor tiene que ser más pequeño que el mínimo de todas los MSIs de todos los flujos admitidos.

Además, hay que tener en cuenta el tipo de patrón de tráfico. Hay que planificar de forma eficiente y simultáneamente tanto el tráfico de tasa de bit constante (CBR), como el tráfico de tasa de bit variable (VBR). El tráfico CBR tiene un patrón de llegada regular, mientras que el tráfico VBR puede variar bastante su patrón de llegada, con un pico de tasa de bit que normalmente es mucho mayor que la tasa promedio. Por tanto, el tráfico VBR es más complicado de caracterizar y planificar que el CBR:

- Se garantiza una tasa mínima igual a la tasa promedio, la llegada de ráfagas grandes provoca retardo alto y la necesidad de colas de tamaño grande.
- Si reserva una tasa igual al pico de la ráfaga, esto provocaría una sustancial pérdida de ancho de banda.

En la literatura disponible hay propuestos varios algoritmos de planificación para HCCA. Algunos se centran en tráfico CBR y planifican ineficazmente el tráfico VBR. Un ejemplo

de algoritmo que no es eficaz para VBR es el algoritmo de planificación básico propuesto, que no ajusta dinámicamente las oportunidades de transmisión (TXOPs) para acomodarse al tráfico actual, sino que las TXOPs son tratadas de manera estática.

Sin embargo, existen otros algoritmos, entre los que se encuentra WTTP (*Wireless Timed Token Protocol*) que ajustan dinámicamente la duración de la TXOP, aunque normalmente requieren una mayor carga computacional. De esta forma, este tipo de algoritmos se adaptan a las condiciones cambiantes, como las ráfagas y picos máximos de las tasas características de tráficos VBR.

Existen dos modelos generales para proveer QoS a una determinada aplicación:

- QoS parametrizada (e.g. IntServ): se basa en garantizar requisitos estrictos de QoS, expresados en términos de valores cuantitativos, tales como retardo, MSI,... Es decir, especificado su TSPEC, se espera que los servicios de datos de la capa MAC cumplan estos requisitos.
- QoS priorizada (e.g. DiffServ): se expresa en términos de prioridad relativa que será utilizada en la transferencia de datos entre dos estaciones. De esta forma, un flujo tendrá mayor prioridad que otro flujo, y estadísticamente (no con seguridad) recibirá un mejor servicio. En este esquema, los valores de QoS, tales como retardo, *jitter*, tasa de bit pueden variar en la transferencia de las tramas de datos, sin la necesidad de reservar los recursos requeridos al negociar el TSPEC entre la estación y el punto de Acceso (AP).

3.2 Capa MAC IEEE 802.11e

IEEE 802.11e es una ampliación del estándar IEEE 802.11 para proporcionar calidad de servicio en redes inalámbricas. Como se comentó anteriormente, IEEE 802.11 especifica dos funciones de acceso al medio, DCF y PCF, pero este estándar no fue diseñado para diferenciar y dar prioridad a los distintos tipos de tráfico. Las aplicaciones en tiempo real como VoIP requieren que no haya pérdidas de paquetes, ni retardo al entregar los datos. Además, las aplicaciones de video y audio necesitan suficiente ancho de banda para tener servicios de calidad. Por otro lado, e-mail y ftp requieren que las transmisiones estén libres de errores. IEEE 802.11e fue diseñado con la idea de poder soportar todas estas aplicaciones con requisitos diferentes.

Como ya se comentó, en IEEE 802.11e hay dos mecanismos de acceso adicionales: EDCA y HCCA. EDCA fue diseñada para soportar priorización de tráficos, tal como lo hace DiffServ, mientras que HCCA soporta tráfico parametrizado (garantizando QoS), al igual que hace IntServ.

Para asegurar la compatibilidad de redes inalámbricas con QoS y sin QoS el estándar permite la existencia conjunta de DCF y PCF con HCF como se puede observar en la figura 3.2:

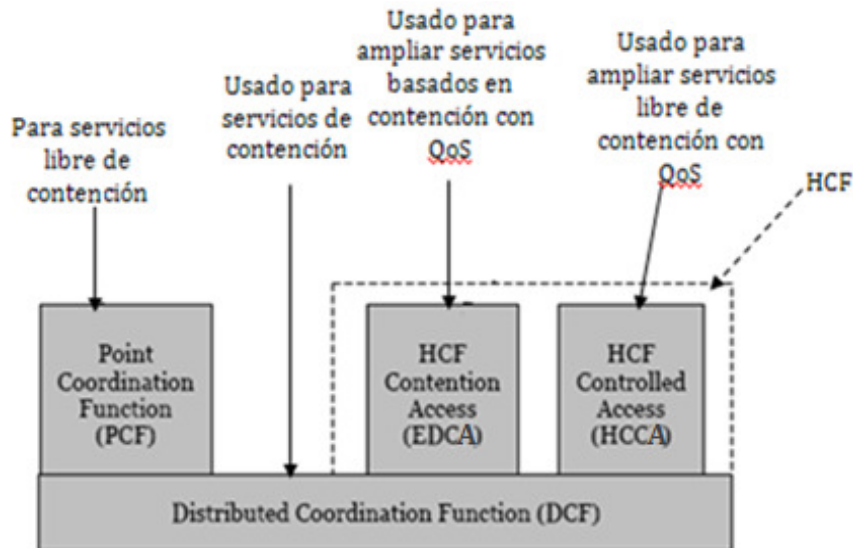


FIG3.2: FUNCIONES DE ACCESO AL CANAL IEEE 802.11 Y IEEE 802.11E

Para saber cuándo utilizar una función u otra, se utiliza un identificador de tráfico (TID) que se incluye en las tramas de la capa MAC. Este identificador tomará un valor entre 0 y 15, según sus requisitos de QoS. Las tramas con valores TID entre 0 y 7 son mapeadas en cuatro colas, siendo cada una de ellas una categoría de acceso (AC). Para acceder a cada una de estas colas se usa el mecanismo EDCA. Por el contrario, las tramas marcadas con un TID entre 8 y 15 son mapeadas en ocho tipos de flujos de tráfico (TS) y usan reglas HCCA. De esta forma, separando la gestión de TSs y de ACs, se consigue QoS parametrizado en los TSs (HCCA) mientras que en las colas AC tenemos tráfico priorizado (EDCA).

3.2.1 EDCA

EDCA proporciona servicios de tráfico diferenciado, es decir, sin reserva de recursos. Por tanto no garantiza que se cumplan los requisitos de QoS estrictamente. Además, la decisión de quién recibe la TXOP se hace de forma distribuida, es decir, se toma conjuntamente entre todas las estaciones. En cada estación con QoS se definen cuatro colas para las ocho tipos de flujos de tráfico existentes, numerados con un TID entre 0 y 7. El esquema de planificación EDCA se muestra en la figura 3.3.

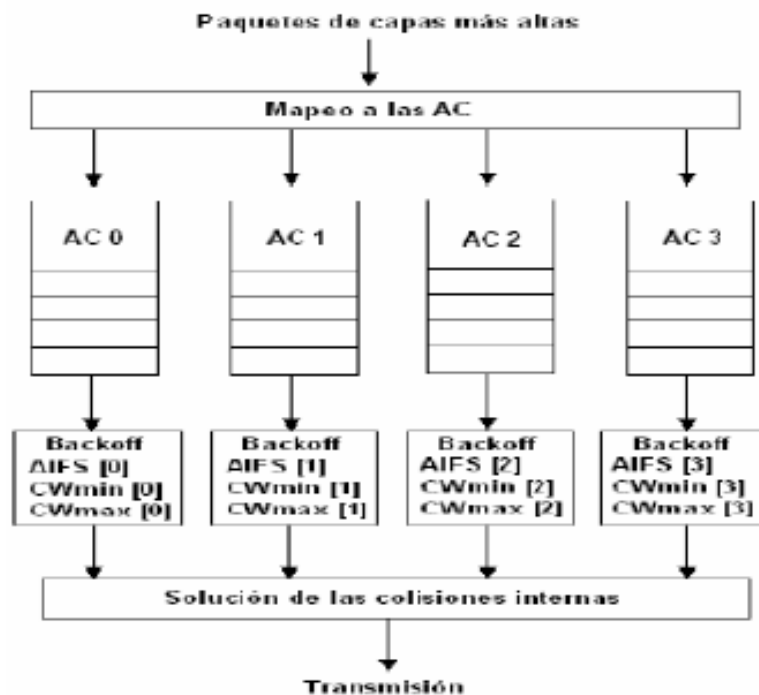


FIG3.3: TRÁFICO EDCA

El mecanismo EDCA, como evolución del IEEE 802.11 DCF, incluye todos los elementos básicos de DCF como el protocolo CSMA/CA, el mecanismo de *backoff* o los distintos tiempos de espacio entre tramas. Aparte de los elementos básicos comunes, añade nuevos componentes que permiten introducir calidad de servicio en el sistema, como son el concepto de TXOP y el AIFS (Espacio entre Tramas Arbitrario). La priorización se consigue en base al uso de los siguientes parámetros (véase la figura 3.4):

- Espacio Arbitrario entre Tramas (AIFS): Su tarea es equivalente al intervalo DIFS usado en DCF, sólo que ahora puede haber un valor diferente de AIFS por cada categoría de acceso, dando una mayor prioridad a las ACs con menor AIFS. El tiempo AIFS es el intervalo de tiempo mínimo que tiene que estar el medio libre antes de empezar a transmitir. Hay que indicar que el AIFS es mayor o igual que DIFS, y por tanto las estaciones con el estándar IEEE 802.11 tiene prioridad sobre EDCA. El valor del AIFS se calcula mediante la siguiente expresión:

$$AIFS[AC] = AIFSN[AC] * aSlotTime + aSIFSTime \quad (4)$$

Donde: *aSlotTime* y *aSIFSTime* representan parámetros conocidos de DCF correspondientes al tiempo de una ranura y el intervalo más corto entre tramas, *AIFSN[AC]* es el denominado número de espacio entre tramas arbitrario y determina la prioridad. La prioridad más alta corresponde al valor más pequeño.

- Ventana de Contención (CW): un número aleatorio para lanzar el mecanismo de espera aleatorio (backoff). Depende de cada estándar y es

distinto para cada AC, tratándose de una prioridad estadística no garantizada.

- Límite de la Oportunidad de Transmisión (límite TXOP): es la duración máxima de la oportunidad de transmisión, durante la cual una QSTA puede transmitir sus paquetes.

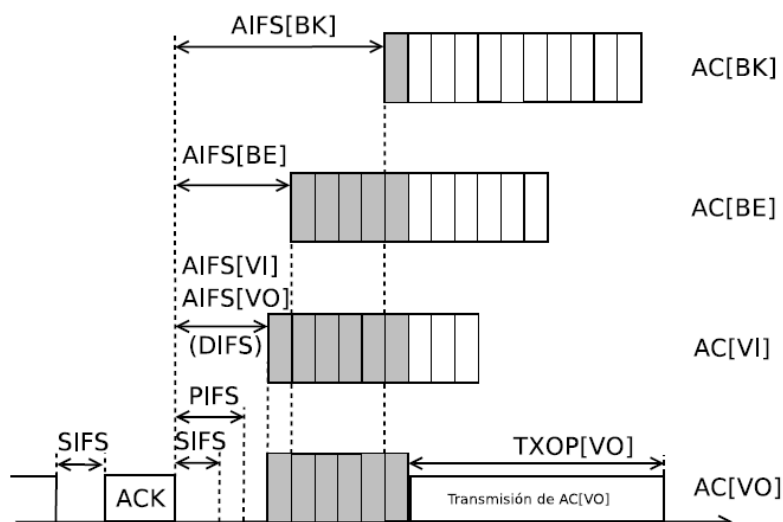


FIG3.4: PARÁMETROS EDCA

Cada AC está asociado a un determinado tipo de tráfico (véase la tabla 3.1). El nombre que se da a cada AC indica el tipo de aplicación en la cola: voz (VO), video (VI), *best-effort* (BE) y *background* (BK).

Prioridad de usuario (UP)	Categoría de Acceso 802.11e	Tipo de servicio
2	0	<i>Best-Effort</i>
1	0	<i>Best-Effort</i>
0	0	<i>Best-Effort</i>
3	1	Prueba video
4	2	<i>Video</i>
5	2	<i>Video</i>
6	3	<i>Voice</i>
7	3	<i>Voice</i>

TABLA 3.1: MAPA DE PRIORIDADES DE USUARIO A CATEGORÍAS DE ACCESO (AC)

Además, cada categoría de acceso dispone de su propia cola de transmisión caracterizada por sus parámetros de acceso: AIFS (Espacio entre Tramas Arbitrario), la Ventana de Contención (CW) y la TXOP, que toman los valores por defecto de las tablas 3.2, 3.3 y 3.4 y que dependerá el estándar en cuestión:

802.11b	AIFSN[AC]	CWmin (SlotTime)	CWmax (Slot Time)	TXOP (μ s)
AC0	2	2	15	3274
AC1	2	15	31	6016
AC2	3	31	1023	0*
AC3	7	31	1023	0*

*El valor 0 indica que sólo se envía un paquete cuando se accede al medio

TABLA 3.2: CONFIGURACIÓN ESTÁNDAR PARA IEEE 802.11B

802.11g	AIFSN[AC]	CWmin (SlotTime)	CWmax (SlotTime)	TXOP (μ s)
AC0	2	3	7	1504
AC1	2	7	15	3008
AC2	3	15	1023	0*
AC3	7	15	1023	0*

*El valor 0 indica que sólo se envía un paquete cuando se accede al medio

TABLA 3.3: CONFIGURACIÓN ESTÁNDAR PARA IEEE 802.11G

802.11e	AIFSN[AC]	CWmin (SlotTime)	CWmax (SlotTime)	TXOP (μ s)
AC0	0	8	16	3000
AC1	0	16	32	6000
AC2	1	32	1024	3000
AC3	5	32	1024	0*

*El valor 0 indica que sólo se envía un paquete cuando se accede al medio

TABLA 3.4: CONFIGURACIÓN ESTÁNDAR PARA IEEE 802.11E

En el caso que una estación opere simultáneamente con distintas AC y el tiempo aleatorio de *backoff* termine en el mismo instante, la categoría con la prioridad más alta es la que empieza a transmitir antes, mientras que el resto se comporta como si se hubiera producido una colisión al acceder al medio. A este tipo de colisión se le denomina colisión virtual.

Vistos los elementos básicos que intervienen en la función EDCA, pasamos a describir su funcionamiento.

En primer lugar, cuando los datos llegan a la capa MAC de 802.11e, ésta se encarga de clasificar adecuadamente los datos y envía la MSDU a la cola correspondiente. Con posterioridad, las MSDUs de las diferentes colas (AC) compiten internamente por la EDCA-TXOP. A partir de aquí, pueden ocurrir tres situaciones distintas:

- El medio está libre y hay una sola cola con información para transmitir: En este caso la EDCA-TXOP es adjudicada a esta cola y podrá transmitir la información almacenada.
- El medio está ocupado: El algoritmo de contención interno calcula, para cada cola (AC), el tiempo de espera aleatorio (*backoff*) y el AIFS. Posteriormente, espera un tiempo igual a la suma del AIFS más el tiempo de espera aleatorio. Tras pasar este tiempo, volverá a comprobar si el medio sigue ocupado. A partir de aquí, si el medio se encuentra libre, empezará a transmitir. Si por el contrario, el medio siguiese ocupado, comenzaría de nuevo el algoritmo de contención interno del AC. El mecanismo de espera es similar al de DCF (AIFS + tiempo aleatorio de *backoff*), y la cola con menor tiempo aleatorio de *backoff* ganará la competición interna.

La cola (AC) vencedora competirá externamente por el acceso al medio inalámbrico. El algoritmo de contención para el medio inalámbrico no se ha modificado significativamente respecto a DCF, excepto que en DCF el rango del tiempo aleatorio de *backoff* y los tiempos de espera son fijos, mientras que en 802.11e son variables y toman valores distintos según la cola (AC) y el estándar correspondiente.

A través de una configuración adecuada de los parámetros de las colas (AC), se puede ajustar el rendimiento del tráfico y se puede lograr la priorización de tráfico.

3.2.2 HCCA

Además de la función distribuida EDCA, el estándar IEEE 802.11e define la función HCCA, que es un mecanismo de acceso al medio centralizado. El coordinador híbrido utiliza esta función para acceder al medio. Normalmente el coordinador híbrido se encuentra en el QAP y puede tomar el control del medio cuando la función lo requiere. De esta forma, el planificador puede asignar las TXOPs a los flujos de tráfico en el momento en que éste decida.

El HC accede al medio después de detectar que el canal está libre durante un tiempo PIFS, que es inferior que un espacio entre tramas DIFS. Por tanto, tiene prioridad sobre el tráfico DCF. Además, el intervalo PIFS es inferior que cualquier espacio entre tramas AIFS, lo que implica que tiene prioridad sobre EDCA para acceder al medio. De esta forma, HCF puede comenzar el acceso al canal en los intervalos CFP y CP indistintamente, mientras que PCF sólo podía acceder al canal en el intervalo libre de contención. Se denomina Período de Acceso Controlado (CAP) al período en el que HCF accede al canal en un intervalo de contención. Consiste en uno o más TXOPs, durante los cuales el punto de acceso puede transmitir SDUs que pertenecen a un flujo descendente o enviar un mensaje de *polling* a una o más estaciones, especificando el tiempo máximo durante el que pueden ocupar el medio.

En el estándar IEEE 802.11 no está limitado el tiempo que una estación puede estar enviando tramas. En HCCA es diferente, ya que el HC indica a cada estación la duración máxima de la TXOP. Esto hace posible que el retardo máximo esté acotado, a diferencia de PCF donde esto no sucedía.

Durante una TXOP, sólo una QSTA accede al medio. Hay dos tipos de oportunidad de transmisión HCCA-TXOP dependiendo del tipo de flujo en cuestión (flujo ascendente y flujo descendente):

- Una TXOP descendente para la transmisión desde un QAP a una QSTA.
- Una TXOP ascendente para la transmisión desde una QSTA a un QAP. Éste se inicia cuando un QAP le envía un mensaje de *polling* a una determinada QSTA. Este mensaje le indica a la QSTA que puede acceder al medio durante un tiempo limitado (límite de la TXOP). Si un flujo de tráfico de una determinada QSTA recibe un mensaje de *polling* pero no tiene tramas para transmitir, la QSTA envía una trama de respuesta al QAP indicándole esta situación y el QAP mandará un mensaje de *polling* a la siguiente estación registrada. El mismo intercambio de mensajes se produce cuando una QSTA determina que el tiempo que tiene disponible en su TXOP es insuficiente para poder transmitir el próximo paquete de su cola.

Otro aspecto importante de la función HCCA es el control de admisión. Una QSTA pide la admisión de un nuevo flujo de tráfico *i* enviando un mensaje ADDTS al HC, que contiene el TSPEC con, entre otros, los siguientes parámetros:

- Tasa de datos promedio (ρ_i)
- Tamaño nominal de la MSDU (N_i)
- Tasa mínima en la capa física (R_i)
- Retardo (D_i)
- Máximo intervalo de servicio (MSI_i)

Los cuatro primeros parámetros caracterizan el flujo de tráfico. Por ejemplo, el retardo especifica el tiempo de vida máximo de una SDU en la estación; El MSI corresponde al tiempo máximo que puede pasar entre dos mensajes de *polling* consecutivos para la misma QSTA.

Para explicar el funcionamiento del control de admisión, se van a describir los parámetros de transmisión de las tramas HCCA. La representación gráfica de estos parámetros la podemos observar en la figura 3.5:

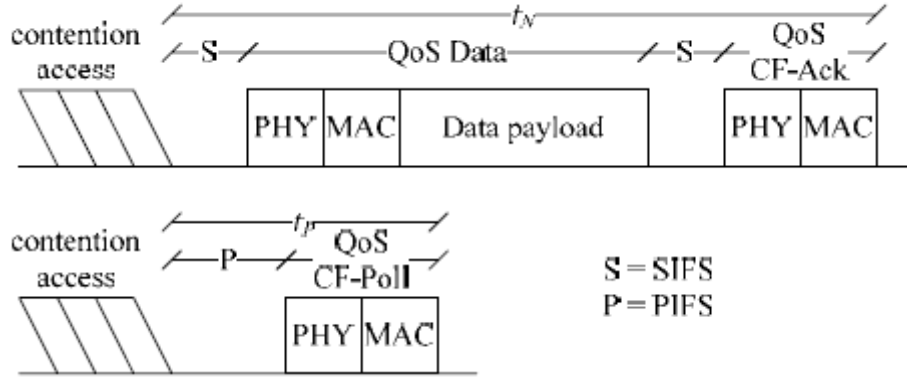


FIG3.5: TRÁFICO HCCA

Se denomina el tiempo de transmisión nominal para el flujo de tráfico i como t_{Ni} es decir, el tiempo necesario para poder transmitir una SDU que pertenece al TS i , incluyendo el tiempo que transcurre para recibir el correspondiente reconocimiento y el espacio entre tramas.

$$t_{Ni} = SIFS + \left(t_{PHY} + \frac{h_{MAC} + N_i}{\Gamma_{DATAi}} \right) + SIFS + \left(t_{PHY} + \frac{h_{ACK}}{\Gamma_{ACKi}} \right) \quad (5)$$

Donde t_{PHY} es el tiempo que dura la cabecera de la capa física, h_{MAC} (h_{ACK}) es el tamaño de la cabecera MAC (trama ACK), y Γ_{DATAi} (Γ_{ACKi}) es la tasa en la capa física para las tramas de datos QoS (ACK).

El tiempo de transmisión de un QoS CF-Poll, t_{pi} incluyendo el espacio entre tramas se puede obtener:

$$t_p = PIFS + \left(t_{PHY} + \frac{h_{POLL}}{\Gamma_{POLL}} \right) \quad (6)$$

Donde h_{POLL} es el tamaño de la trama QoS CF-Poll y Γ_{POLL} es su tasa en la capa física. La sobrecarga debido a los mensajes *polling* de TS i , t_{pi} es igual a t_p si i es un TS ascendente. En otro caso $t_{pi}=0$.

Según el estándar, el QAP es responsable de planificar las oportunidades de transmisión de tal manera que se cumpla el TSPEC negociado con cada uno de los flujos de tráfico admitidos. Concretamente, para cualquier intervalo de tiempo $[t_1, t_2]$ mayor que el Intervalo Especificado Mínimo, que es anunciado por el QAP, la función HCCA se encarga de dar al flujo del tráfico i un número de TXOPs cuya duración acumulada $W_i(t_1, t_2)$, expresión obtenida de [8]:

$$W_i(t_1, t_2) \geq \left\lceil \frac{R_i}{N_i} (t_2 - t_1 - \Delta_i) \right\rceil * t_{N_i} \quad (7)$$

Donde Δ_i es igual al MSI_i , si este se especifica en el TSPEC, o en otro caso es igual a D_i . t_{N_i} es el tiempo que tarda en transmitirse un paquete a la tasa de la capa física.

Básicamente la ecuación anterior implica que el planificador tiene un tiempo suficiente para poder transmitir en $[t_1, t_2]$ tantos bits como correspondan asignar (según el TSPEC) a la tasa R_i , sabiendo que la velocidad real es Γ_i .

Por último, aunque el documento 802.11e no especifica un planificador concreto para el QAP, se ha propuesto un algoritmo básico para propósitos informativos. El planificador de referencia utiliza el algoritmo *round-robin*. Para ello, sirve a cada TS de forma periódica, donde el período (llamado intervalo de servicio, SI) es el mismo para todos los flujos de tráfico. Éste intervalo de servicio toma el valor más pequeño de Δ_i . La duración de la TXOP del TS i se pone al valor $\lceil (R_i * SI) / N_i \rceil * t_{N_i} + t_{p_i}$, de forma que se cumple la desigualdad anterior.

Supertrama IEEE 802.11e

Una vez visto EDCA y HCCA, para finalizar se va a describir la supertrama IEEE 802.11e:

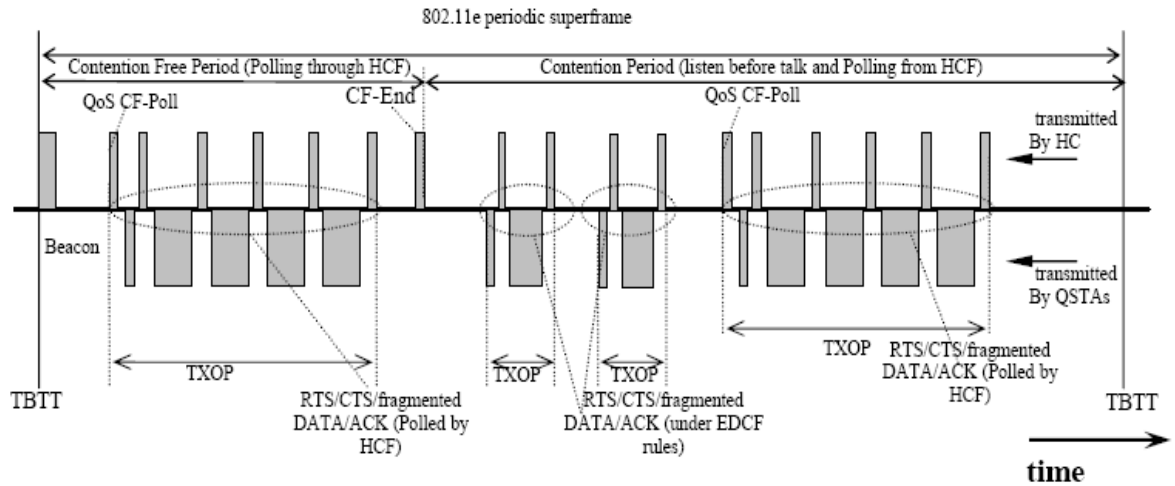


FIG3.6: SUPETRAMA IEEE802.11E

Igual que la supertrama IEEE 802.11, una supertrama IEEE 802.11e consiste en dos períodos:

- Período Libre de Contención: período de la supertrama en la que el acceso al medio es indicado por el HC. Las estaciones sólo aceptan o rechazan las peticiones de transmisión. El inicio de éste período se indica en las tramas

QoS CF-Poll y finaliza en el tiempo anunciado en la trama *beacon*. También puede terminar por el envío de una trama CF-End del HC.

- Período de contención: período de la supertrama en la que se usan algoritmos de contención para acceder al canal. El inicio de la TXOP se consigue usando reglas EDCA (tras un periodo AIFS + un intervalo de *backoff*), o bien cuando la QSTA recibe una trama QoS CF-Poll del HC (en el intervalo HCCA dentro del período de contención).

3.3 Algoritmos para IEEE 802.11e HCCA

Son varios los algoritmos que existen para proporcionar QoS en redes WLAN. En concreto, y dado que en este proyecto se ha usado HCCA, se van a describir los principales algoritmos existentes en la literatura para la función de coordinación centralizada:

En la literatura actual se pueden distinguir tres categorías fundamentales:

- Algoritmos basados en tiempo de expiración de paquete: algoritmos bien estudiados y ampliamente usados en la planificación de sistemas operativos multiprogramados. IEEE 802.11e también usa estos algoritmos para planificar el reparto y duración de los turnos.
- Algoritmos basados en la estimación de la longitud de la cola: adaptan la TXOP a la carga actual de cada una de las colas, y sus requisitos.
- Algoritmos híbridos HCCA/EDCA: estiman la carga del canal y posteriormente usan servicios diferenciados basados en EDCA para repartir las TXOPs. Se usa HCCA, para los flujos que necesitan garantizar sus requisitos de QoS.

En esta sección se describen las características generales de cada una de estas categorías, se explicarán algunos algoritmos representativos de cada uno de estos grupos. Por último, se expondrán sus ventajas e inconvenientes y la motivación de la selección del planificador WTTT como algoritmo para desarrollar el proyecto.

IEEE 802.11 Task Group E (TGe) se creó para mejorar la capa MAC del estándar IEEE802.11 y de esta forma, como ya se comentó anteriormente, hacer posible que las redes WLAN soporten QoS. El principal objetivo alcanzado por el TGe, es un algoritmo de planificación que es tomado como referencia. Se trata de un algoritmo basado en multiplexación temporal (TDM):

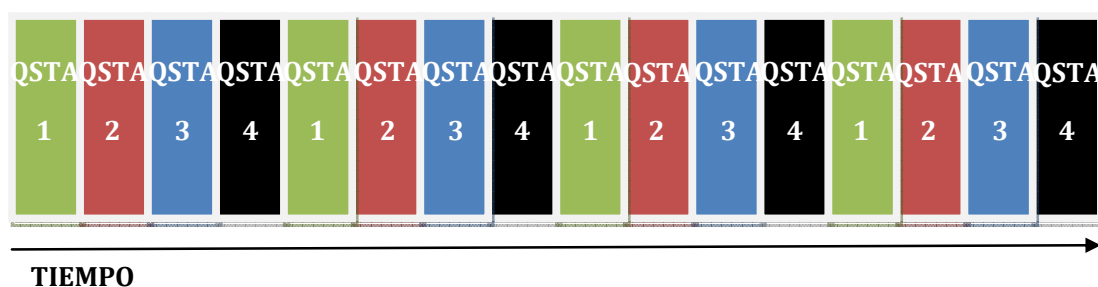


FIG3.7: REPARTO DE LAS TXOPS DEL ALGORITMO BÁSICO DE IEEE 802.11E

En la figura 3.7 se observa un ejemplo de la distribución de los turnos entre las estaciones. Cada una de las estaciones dispone de un tiempo fijo para transmitir. Este tiempo es el mismo para todas las estaciones. Además, el acceso al canal se va repartiendo por turnos, siguiendo el mismo orden (en la figura 3.7, el orden es QSTA1, QSTA2, QSTA3, QSTA4, QSTA1, ...). Todas las estaciones tienen alguna oportunidad para transmitir antes de que su intervalo de servicio mínimo especificado en el TSPEC expire. Si un flujo de una estación no tiene tramas para transmitir, llegado su turno, se le asignaría igualmente una TXOP. Una vez transcurrido el tiempo asignado a la estación, la TXOP pasa a la siguiente estación.

Este algoritmo no es adecuado para aplicaciones VBR, ya que las TXOPs y el orden en el que se reparten los turnos son fijos, y no se adaptan a las características actuales del tráfico, que pueden ser variables.

De esta forma, a cada flujo de tráfico se le asigna una capacidad fija que además será periódica. El período entre dos asignaciones consecutivas se conoce como intervalo de servicio (véase SI en la figura 3.8).



FIG3.8: INTERVALO DE SERVICIO DEL ALGORITMO IEEE 802.11E

Este SI es el mismo para todos los flujos que han sido admitidos. Se calcula como el mínimo mSI del TSPEC de todos los flujos admitidos. En el caso de que no esté especificado el mSI, se tomará el valor más pequeño de entre todos los retardos de los TSPEC admitidos:

$$SI = \min_i \{D_i, MSI_i\} \quad (8)$$

Por otro lado, la duración de la TXOP tomará el menor valor que cumple la expresión 7.

IEEE 802.11e además requiere que cualquier algoritmo implementado para proporcionar QoS en HCCA cumpla con esta expresión. La TXOP tiene que ser redondeado superiormente para que puedan transmitirse un número entero de paquetes de tamaño SDU nominal. Es decir, se le asigna a cada flujo una TXOP suficiente para que el número de paquetes que entren en esa oportunidad de transmisión sea un número entero. Además hay que tener cuidado con el *head of line blocking*, ya que si el paquete a transmitir es de gran tamaño, y la TXOP calculada es más pequeña, podría producir un bloqueo permanente no deseado. Para evitar esta situación, el cálculo de la TXOP elige un máximo entre el cálculo anterior y el tiempo necesario para transmitir un paquete de tamaño máximo:

$$TXOP_i = tx(P).I(i) + \max \left\{ \left\lceil \frac{R_i \cdot SI}{N_i} \right\rceil \cdot t_{Ni}, tx(M_i, \Gamma_i) \right\} \quad (9)$$

Donde $tx(P)$ es el tiempo para transmitir una trama de *polling* e $I(i)$ es una función indicadora que vale 1 cuando i es un TS ascendente, y 0 si es descendente.

Dado que el algoritmo de referencia tiene la TXOP y el intervalo de servicio fijos, este algoritmo no es adecuado para tráfico VBR, como por ejemplo la transmisión de un video con codificación (e.g. MPEG-4). Esto queda demostrado en [1]. En éste proyecto se ha evaluado el rendimiento del algoritmo de referencia propuesto por el TGe para comprobar si este algoritmo se adapta o no al tráfico cambiante VBR. Se demuestra mediante simulación que para tráfico CBR el retardo se mantiene dentro de los niveles de calidad requeridos, incluso para carga alta. Sin embargo, para VBR, los resultados son poco eficientes en cuanto a los requisitos de QoS se refiere. Esto se debe a que las ráfagas de datos incrementan rápidamente el tamaño de la cola, provocando un aumento del retardo.

En este proyecto, un objetivo es el soporte de tráfico VBR cumpliendo con sus requisitos de QoS, es decir, poder transmitir y recibir tráfico heterogéneo con calidad de servicio. El aumento del ancho de banda disponible, así como la mejora de las prestaciones de los dispositivos inalámbricos han provocado un desarrollo de fuentes de datos con tráfico multimedia. Este tráfico multimedia no comparte los mismos requisitos *best-effort* de una red tradicional. Se necesita soportar mecanismos de QoS para asegurar una calidad mínima de los servicios. Por ejemplo, en la difusión del audio y vídeo (*streaming*) los retardos o el *jitter* que no estén controlados y delimitados provocan una disminución de la calidad.

A continuación, se va a hacer una clasificación de los algoritmos existentes en la literatura. Se pueden distinguir tres grupos:

- Algoritmos basados en el tiempo de expiración de paquete.

- Algoritmos basados en la estimación de la longitud de la cola.
- Algoritmos híbridos HCCA/EDCA.

3.3.1 Algoritmos basados en el tiempo de expiración de paquete

El problema de planificar en tiempo real es un tema ampliamente estudiado en sistemas operativos multiprogramados. Los algoritmos basados en la finalización de paquetes tratan de adaptar los algoritmos de sistemas operativos multiprogramados existentes al campo de las comunicaciones. Una de las ventajas de estos algoritmos es que ya están bien estudiados y comprobada su eficiencia, y por tanto sólo se trata de aplicarlo a otra disciplina.

A continuación se describen los algoritmos RTH, SETT-EDD y el Algoritmo de Korakis y Tassiulas como ejemplos representativos de esta categoría de planificador.

3.3.1.1 RTH

RTH (HCCA para Tiempo Real) pertenece al conjunto de algoritmos basados en el tiempo de expiración del paquete. Se trata de un algoritmo de planificación para la función HCCA del estándar IEEE 802.11e. En cada TXOP, el algoritmo RTH da permiso a un flujo determinado para transmitir una cantidad fija de la capacidad total del canal. RTH tiene como objetivo minimizar la sobrecarga causado por las tramas de *polling*, manteniendo una complejidad computacional práctica y razonable.

En las simulaciones realizadas en [2], se observa que el algoritmo RTH asigna mayor capacidad a las estaciones que usan EDCA que el algoritmo de referencia, manteniendo el cumplimiento de los requisitos del tráfico transmitido mediante HCCA. Además, con los mismos requisitos, se pueden admitir más estaciones con el algoritmo RTH que con el algoritmo básico.

El planificador RTH está compuesto por cuatro bloques, como se puede observar en la figura 3.9:

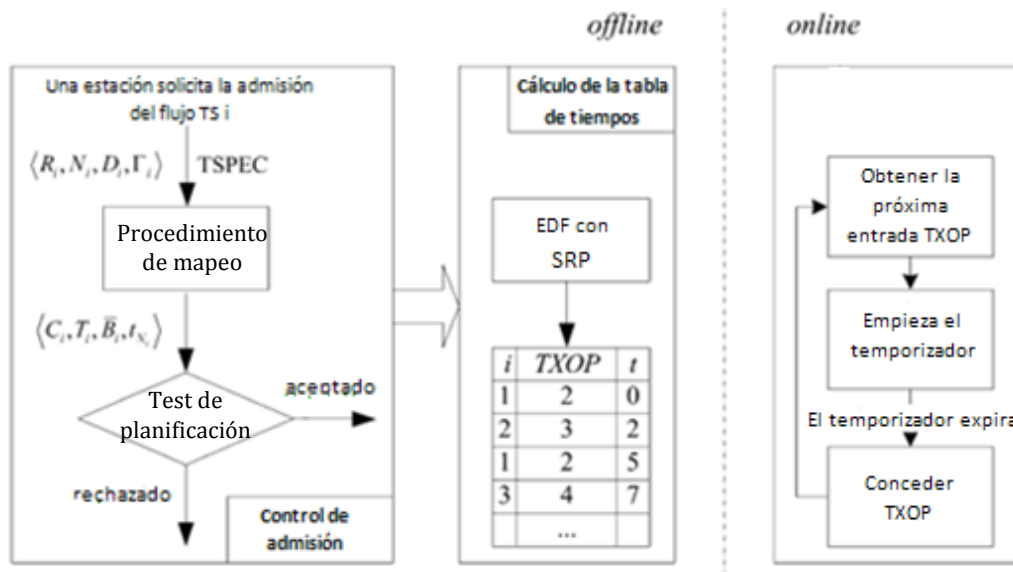


FIG3.9: ELEMENTOS DEL PLANIFICADOR RTH

- Procedimiento de mapeo: en este bloque se calculan una serie de parámetros necesarios para desarrollar el algoritmo. Para el cálculo de estos parámetros se usa el TSPEC.
- Test de planificación: en este bloque se realiza el control de admisión.
- Procedimiento *Offline*: en este bloque se calcula una tabla de tiempos periódica de TXOPs, que es usada como entrada para el procedimiento *online*.
- Procedimiento *Online*: esta parte concede dinámicamente las TXOPs a los TSs admitidos.

RTH combina el algoritmo "primero el más urgente" (EDF), con el protocolo de la pila de recursos (SRP). EDF asume que las tareas se van generando arbitrariamente. Ya que este no es el caso cuando se están planificando TXOPs, se introduce el concepto de bloqueo con SRP. Una tarea puede ser bloqueada durante un tiempo limitado por otra, con un tiempo de expiración de paquete posterior, si ésta está en una sección crítica. Todas las tramas transmitidas bajo IEEE 802.11e son consideradas secciones críticas. Es decir, si una QSTA accede al medio, no podría acceder ninguna otra.

3.3.1.2 SETT-EDD

La Planificación Basada en Tiempos Estimados de Transmisión– con Fecha de Cumplimiento más Temprana (SETT-EDD) es una estrategia de planificación incluido en el grupo de los algoritmos basados en el tiempo de expiración del paquete. Trata de mejorar la eficiencia del algoritmo básico para patrones de tráfico VBR, como MPEG-4, que tienen tamaños de paquete distintos y ráfagas de datos. Con el algoritmo básico,

las ráfagas de datos se van transmitiendo en varios TXOPs, separados por espacios de tiempo SI, lo que provoca un retardo significativo de la transmisión.

SETT-EDD necesita que los flujos de tráfico especifiquen un parámetro adicional del TSPEC, el parámetro mSI , que como se comentó es el tiempo mínimo entre dos intervalos de servicio consecutivos. Consiste en hacer más flexibles las TXOPs de cada una de las estaciones. Se le asigna a cada SDU un tiempo para empezar su transmisión t' igual al tiempo de llegada del mensaje de *polling* más el valor de mSI , y un tiempo de finalización igual al tiempo de llegada del *polling* más el MSI, es decir:

$$t+mSI \leq t' \leq t+MSI \quad (15)$$

Esta expresión $t+mSI$ indica el instante de tiempo menor en el cuál el siguiente mensaje de *polling* debe ser recibido. Por otro lado, $t+MSI$ es el tiempo máximo en el que el siguiente mensaje de *polling* tiene que ser recibido por una estación.

El HC tiene que decidir a qué estación le manda el primer mensaje de *polling*, entre todas las que satisfacen la ecuación 15 en un momento dado. Como todas las estaciones transmiten en el mismo medio inalámbrico en un instante de tiempo distinto, este tipo de planificación es similar a la utilizada en sistemas multiprogramados en tiempo real. El algoritmo Fecha de Cumplimiento más Temprana (EDD) es muy eficiente en problemas de planificación en tiempo real. Debido a esto y a su simplicidad, se usa con SETT. Concretamente se usa EDD con retardo, que es una variante de EDD y calcula los tiempos de expiración del paquete a partir del retardo. El tiempo de expiración del paquete en SETT-EDD es $t+MSI$. El control de admisión es igual que el del algoritmo de referencia de IEEE 802.11e.

3.3.1.3 Algoritmo de Korakis y Tassiulas

Este algoritmo tiene en cuenta que existe tráfico CBR y VBR. El algoritmo se ejecuta periódicamente por el HC, que sólo considera las QSTAs que detectan un buen estado del canal para lo que se requiere un protocolo de señalización en la capa MAC que notifique a la QAP el estado del canal en cada instante.

Los flujos con tráfico CBR se sirven primero, siguiendo el algoritmo de referencia de IEEE 802.11e. El resto del intervalo de servicio se planifica para tráfico VBR según un mecanismo de tiempo límite (*deadline*): una TXOP se sirve tan tarde como sea posible sin sobrepasar dicho tiempo, que toma el valor del retardo máximo permitido. Ya que el QAP no sabe el tiempo de llegada del TS ascendente, se toma el peor caso: el siguiente paquete se encola tan pronto como el paquete anterior ha sido enviado. Aunque el número de operaciones requeridas para planificar un solo TXOP es bajo, el cálculo de la planificación periódica aumenta su complejidad, debido a la necesidad de ordenar las TXOPs basándose en su tiempo límite. Para no perder capacidad en entornos reales, el QAP puede ser forzado a calcular la planificación mucho antes del comienzo del intervalo de servicio, por tanto usando información incompleta.

3.3.2 Algoritmos basados en estimación de la longitud de la cola

Otra posibilidad para trabajar con tráfico ascendente VBR es estimar la longitud de las colas de los flujos ascendentes, y de esta forma adecuar la longitud de las TXOPs al tráfico existente.

Esto se puede realizar utilizando el mecanismo opcional de *piggybacking* del tamaño de las colas. Concretamente, en el Planificador Dinámico Basado en Realimentación (FDBS) la información entregada por la QSTA sobre el estado de sus colas se utiliza como entrada en un sistema de control en el QAP, que estima los requisitos de los TSs. Por tanto, la duración de la TXOP depende de los requisitos estimados por el sistema de control durante cada período. Si la suma de todas las asignaciones de TXOPs excede el intervalo de servicio, entonces todas las TXOPs se reducen proporcionalmente.

Por otro lado, HCF justo (FHCF), utiliza en el promedio de las muestras de longitud de las colas recogidas durante una ventana de tiempo de tamaño fijo. La duración de las TXOPs se calcula proporcionalmente a las distribuciones de la longitud de las colas. Antes de conceder las TXOPs a las estaciones, el HC redistribuye de manera proporcional el intervalo de servicio que queda sin usar, si existe, o disminuye la duración de las TXOPs de manera proporcional si la suma de todas las TXOPs excede el intervalo de servicio.

3.3.3 Algoritmos híbridos HCCA/EDCA

Finalmente, se van a revisar los algoritmos basados en el uso de EDCA, ya sea solo o en combinación con HCCA, para dar a los flujos de tráfico QoS parametrizado. En este tipo de mecanismos, HCCA sirve tráficos con requisitos de calidad de servicio de tal forma que queden garantizados. Se usa EDCA para tráfico TCP.

En el algoritmo ChaPlet en [11], los autores proponen permitir al tráfico VBR acceder al medio usando tanto HCCA como EDCA. Para ello se hace una estimación de la carga del canal y se va haciendo un uso dinámico de HCCA y EDCA, según las condiciones lo requieran. Si las estimaciones del canal indican que la carga del canal es baja, se usa EDCA. Sin embargo, si las estimaciones del canal reflejan que se puede utilizar HCCA, éste toma el control del canal. Para comprobar el estado del canal, se mide el coeficiente de utilización del canal y el número de colisiones.

3.4 Conclusión

Después de estudiar algunos algoritmos representativos en la literatura, y compararlos en distintos aspectos como la facilidad de implementación, complejidad computacional, eficiencia con tráfico VBR, eficiencia con EDCA, se ha llegado a la conclusión de que *Wireless Timed Token Protocol* (WTTTP, véase el apartado 4) es un algoritmo que cumple con eficiencia todos estos requisitos. Tiene menor coste de hardware y tiene mayor rendimiento (tasa de bit alcanzada, *jitter*, retardo,...) en aplicaciones con necesidad de transmitir a tasas de bit altas.

RTH permite que TSs con diferentes requisitos sean planificados eficientemente y, por tanto, admite más flujos que el algoritmo de referencia de IEEE 802.11e. Sin embargo, RTH no soporta tráfico VBR.

SETT-EDD tiene también un rendimiento superior al algoritmo básico, especialmente con tráfico VBR. Sin embargo muestra gran complejidad computacional

El algoritmo de Korakis y Tassiulas genera mayor complejidad computacional debido a que ordena las TXOPs según el tiempo de finalización de los paquetes.

Los algoritmos basados en estimación de la longitud de la cola requieren una alta carga computacional al principio de cada intervalo de servicio, debido a la estimación de la longitud de la cola y el ajuste de las TXOPs.

Por último, en los algoritmos híbridos HCCA/EDCA, el acceso aleatorio al medio de EDCA, hace imposible poder determinar un retardo garantizado a priori.

De esta forma, el algoritmo elegido es WTPP. A continuación se describe con detalle este algoritmo, incluyendo la simulación de varios escenarios y una comparativa con el algoritmo básico.

4. WTTT

Wireless Timed Token Protocol es otro ejemplo de planificador para el mecanismo de acceso al medio HCCA del estándar IEEE.802.11e. WTTT [5] es capaz de reservar una tasa mínima a cada una de las estaciones y soporta tanto tráfico de tasa de bit constante (CBR) como de tasa de bit variable (VBR) con requisitos de QoS. También puede gestionar tráfico sin requisitos de QoS concretos, como el generado por las aplicaciones que utilizan el protocolo TCP. Es decir, puede trabajar de forma eficaz con cualquier tipo de tráfico, aunque esto implica una mayor carga computacional que el algoritmo de planificación de referencia.

La forma que tiene el protocolo de gestionar estos tres tipos de tráfico se explica a continuación:

- Tasa de bit constante (CBR): se le proporciona una capacidad fija para garantizar la tasa de bit especificada en el TSPEC.
- Tasa de bit variable (VBR): se garantiza una tasa mínima, pero si hay ancho de banda no utilizado, se reparte entre las fuentes VBR para absorber picos de tráfico y reducir su retardo.
- Tráfico sin requisitos de QoS: si después de realizar la reserva para tráficos CBR y VBR queda aún ancho de banda disponible, éste se dedica a la transmisión de tráfico TCP (sin requisitos de QoS concretos). Este tipo de tráfico se transmite usando el mecanismo de acceso EDCA.

El coordinador híbrido planifica las oportunidades de transmisión (TXOP) tanto de las QSTAs (sentido ascendente) como del QAP (sentido descendente). Estas oportunidades de transmisión, al contrario de lo que sucede con el algoritmo de planificación de referencia, son de tamaño variable, por lo que se pueden acomodar al tráfico VBR.

WTTT pertenece al grupo de los algoritmos basados en la tasa servida, en lugar de estar basados en el tiempo de finalización de paquete. Está inspirado en *Timed Token Protocol* (TTP), usado en el protocolo de Interfaz de Datos Distribuidos por Fibra (FDDI). Este algoritmo no requiere tanta carga computacional como otros algoritmos que tratan el tráfico VBR eficientemente (e.g. algoritmos basados en el tiempo de finalización de paquete). Sin embargo, tiene mayor carga computacional que los algoritmos que sólo trabajan de forma eficaz con el tráfico CBR (e.g. el algoritmo de referencia para HCCA).

WTTT se basa en un algoritmo *round robin*, esto es, un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden determinado, comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento. En este sentido es similar al algoritmo de referencia. Ambos tienen dos listas, si bien en el primero están todos los flujos admitidos, y en el segundo para entrar en la lista de adjudicación de TXOPs se deben cumplir algunos requisitos.

La lista *round robin* se compone de nodos (flujos ascendentes o descendentes) y se encuentra en el HC. Además, existe un nodo especial que representa el intervalo de tráfico con acceso basado en contención (véase 2.3.4). Este nodo representa el tráfico sin QoS, transmitido utilizando EDCA.

Cuando se visita un nodo de la lista, nodo que pertenecerá a cualquiera de los tipos comentados anteriormente, el planificador calcula su tiempo de transmisión. Además, planifica una TXOP para el nodo (si el nodo tiene un flujo CBR o VBR) o se abstiene de generar un período de acceso controlado durante el tiempo calculado (si el nodo utiliza acceso con contención, e.g. EDCA o DCF).

A continuación se describe cómo se gestiona la lista *round robin*, y posteriormente cómo se calcula el tiempo de transmisión. Posteriormente se explica el mecanismo de control de admisión y el cálculo adaptativo de los parámetros del algoritmo.

4.1 Gestión de la cola con disciplina *round robin*

La gestión de la cola permite dar a los distintos nodos admitidos la posibilidad de transmitir. Como regla general, los nodos deberán estar en la lista *round robin* sólo si tienen tráfico a enviar. Por esta razón, existen dos tratamientos distintos dependiendo de que el flujo sea ascendente o descendente (véase la figura 4.1):

- Flujos de tráfico descendente:

Estos flujos generan tráfico desde el QAP, que conoce el estado de dichos flujos. Por tanto, se añaden o eliminan de la cola dependiendo de que queden paquetes por transmitir o no.

- Flujos de tráfico ascendente:

Estos flujos generan tráfico desde las estaciones. En este caso, el QAP no conoce el número de paquetes pendientes de ser transmitidos, ya que los paquetes se encuentran en las QSTAs.

Por tanto, la gestión de estos nodos es diferente al de los anteriores. Si se considerase que estos nodos tienen siempre paquetes para transmitir, esta suposición provocaría que se diesen turnos a estaciones que no lo necesitan, y por tanto, se desaprovecharían recursos con mensajes de *polling* y respuestas sin datos.

Una alternativa simple, aunque efectiva, es que cada QSTA comunique al QAP el número de paquetes pendientes de ser transmitidos mediante *piggybacking* (contemplado en el estándar IEEE 802.11). De esta forma, el planificador puede eliminar el nodo de la lista cuando la cola asociada al flujo se anuncia vacía.

Para saber cuándo añadir el nodo de nuevo a la lista *round robin* (en el caso de que no tuviese paquetes para transmitir), se parte de las hipótesis de que muchas aplicaciones, tanto CBR como VBR, generan paquetes en intervalos constantes de tiempo y que además cada QSTA especifica este

intervalo en el TSPEC. Por ello, se utiliza el intervalo de servicio mínimo (mSI del TSPEC) para calcular el momento en el que se debe incluir de nuevo un nodo en la lista, es decir, cuando pasa un intervalo de tiempo igual a su intervalo de servicio mínimo. De esta forma, el QAP supondrá que se ha creado al menos un paquete a la velocidad de generación anunciada.

Esta estrategia reduce las respuestas nulas (tramas sin datos porque no hay ningún paquete en la cola) a un mensaje de *polling*, con el consiguiente ahorro de ancho de banda. Si un nodo responde a un mensaje de *polling* indicando que no tiene mensajes a transmitir, el siguiente nodo de la lista se selecciona inmediatamente.

Por último, el nodo que representa al tráfico que accede con contención (EDCA o DCF) está siempre en la lista *round robin*, ya que el QAP no puede hacer ninguna suposición sobre este tipo de tráfico.

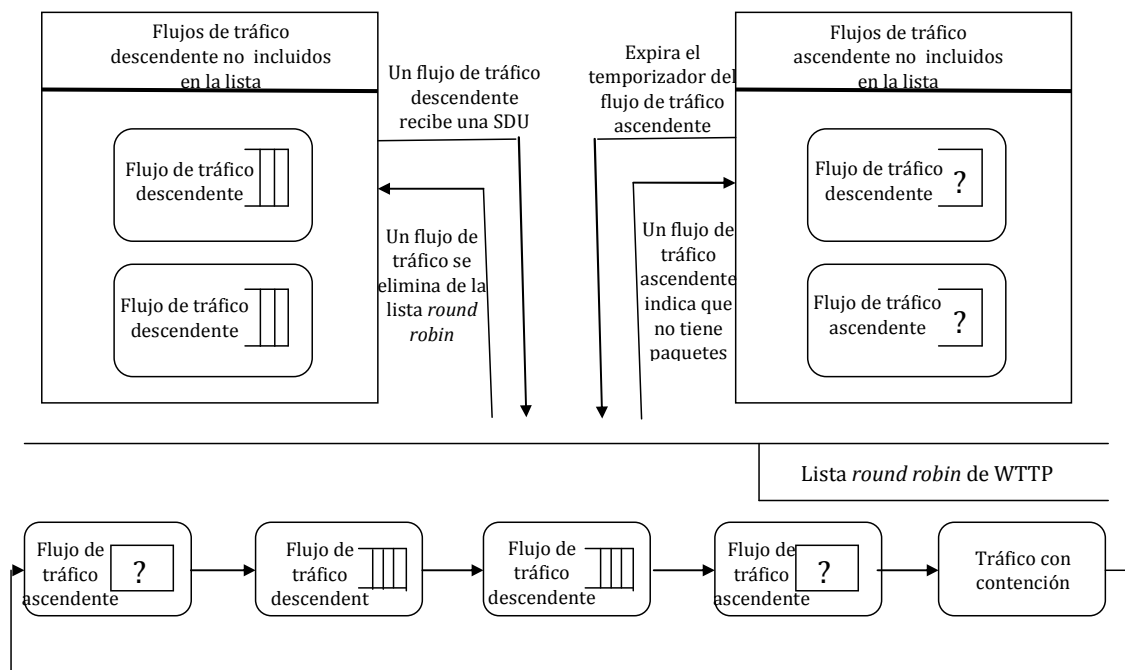


FIG4.1: GESTIÓN COLA CON DISCIPLINA ROUND ROBIN

4.2 Cálculo del tiempo de transmisión

Una vez comprendido el mecanismo de entrada y salida de la lista de cada uno de los tipos de nodos, se va a explicar el cálculo del tiempo de transmisión. Existen varios parámetros importantes en el cálculo de este parámetro:

- *Target Token Rotation Time (TTRT)*: es el tiempo tomado como referencia para la duración de un ciclo completo de la lista.
- *Token Rotation Timer (TRT)*: este temporizador es propio de cada nodo. Mide el tiempo que ha pasado desde la última que el nodo recibió un turno.

El tiempo de transmisión del planificador de un nodo viene determinado por los siguientes valores:

- Ancho de banda síncrono (H_i): tiempo fijo en cada ciclo.
- Ancho de banda asíncrono (α_i): tiempo que varía en cada ciclo.

Al ser elegido el nodo i (una vez cada ciclo, e.g cada intervalo $TTRT$), el QAP le asignará un ancho de banda calculado como suma de una parte fija (ancho de banda síncrono) y otra variable (ancho de banda asíncrono). Para calcular el ancho de banda asíncrono, el temporizador TRT_i se inicializa al valor de $TTRT$, y realiza una cuenta atrás comenzando desde la última vez que el nodo fue servido. Con el valor de TRT_i se calcula el ancho de banda asíncrono usando la expresión 16:

$$\alpha_i = \begin{cases} 0 & TRT_i < 0 \\ \min\{TTRT - H_i, TRT_i\} & TRT_i \geq 0 \end{cases} \quad (16)$$

Esta expresión indica que si el temporizador TRT_i es menor que cero no se dispondrá de ancho de banda asíncrono para transmitir. Esto se debe a que la llegada del siguiente turno de transmisión se recibe en un tiempo superior al $TTRT$ por lo que no queda disponible ancho de banda para la componente asíncrona. En este caso TRT_i se restablece a $TTRT + TRT_i$.

Sin embargo, si el TRT_i es mayor que cero, se reservará un ancho de banda variable para ese nodo. En este caso, el turno para poder transmitir de nuevo se asigna antes de que pase un intervalo $TTRT$ y significa que hay disponible ancho de banda para la componente asíncrona. Una vez asignado el turno al nodo, TRT_i se restablece a $TTRT$.

Hay que tener en cuenta que no todo el ancho de banda asignado a cada uno de los flujos de tráfico en un ciclo (ancho de banda síncrono o asíncrono) será utilizado. En el caso de un flujo de tráfico descendente, si se envían todos los datos a transmitir, el nodo se elimina de la lista *round robin*. En el caso de un flujo de tráfico ascendente, parte de la TXOP se puede quedar sin usar debido a que no se sabe con seguridad si una QSTA tiene paquetes que transmitir. Por otro lado, ya que el QAP no sabe cuántas estaciones desean acceder al medio usando EDCA, la capacidad asignada para este cometido se usa siempre.

4.3 Control de admisión

Para poder cumplir con los objetivos de garantizar la tasa de bit mínima y tener un retardo de paquete acotado, el coordinador híbrido no puede dar servicio a un número ilimitado de fuentes. Si una estación quiere añadir un flujo nuevo, tendrá que cumplir la siguiente condición (véase [5]):

$$\sum_{nodo i} H_i + \tau \leq TTRT \quad (17)$$

Es decir, la suma del ancho de banda síncrono de todos los nodos que han sido admitidos, más un término τ que se explicará posteriormente, es menor o igual que el valor de $TTRT$. El parámetro $TTRT$ corresponde con el tiempo de espera promedio entre servicios para un determinado nodo. Además $2 * TTRT$ es el límite superior del intervalo de servicio, ya que el parámetro $TTRT$ corresponde con la mitad del retardo mínimo especificado en el TSPEC [5]. Con esta condición, un nodo que tiene un ancho de banda síncrono igual a H_i , tiene garantizada una tasa promedio igual a $H_i / TTRT$ veces la capacidad del canal, y tiene un tiempo de acceso al medio (intervalo de servicio) limitado (véase [5]).

El parámetro τ refleja la incertidumbre introducida por los mecanismos de acceso con contención (e.g. EDCA y DCF). En general, el QAP no comienza un período de acceso controlado hasta que el tiempo asignado a la transmisión anterior no ha finalizado. Sin embargo, podría suceder que una estación empiece la transmisión de una trama, usando DCF/EDCA justo antes de que el QAP recupere el control del medio para comenzar una nueva CAP. En este caso, el acceso mediante EDCA/DCF ocupa el medio durante más tiempo de lo esperado. Este tiempo extra es referido como τ y tiene un límite superior dado por el máximo entre:

- La máxima duración de una TXOP-EDCA.
- El tiempo para transmitir la trama con mayor tamaño a la menor tasa de transmisión del medio inalámbrico.

4.4 Cálculo de los parámetros $TTRT$ y H_i

Para el cálculo de estos parámetros se hace uso de los campos de la especificación del tráfico comentados en apartados anteriores. El QAP y la QSTA negocian estos datos en la fase de control de admisión.

A continuación se muestra el cálculo del $TTRT$:

$$TTRT = \frac{1}{2} \min_I D_i \quad (18)$$

Por tanto, para obtener el valor del $TTRT$, se utiliza la mitad del mínimo de entre todos los retardos de los TSPEC de todas las QSTAs. De esta forma, $2 * TTRT$ es el máximo tiempo que puede tardar un nodo en recibir dos turnos de transmisión consecutivos.

$$2TTRT = \min_I D_i \quad (19)$$

La expresión 19 garantiza la tasa de bit especificada en el TSPEC concediendo oportunidades de transmisión a una tasa suficiente para cumplir con esta condición [5]. Sin embargo, no garantiza un retardo acotado (véase [5]) ya que las TXOPs

asignadas a un nodo no tienen por qué ser suficientes para poder transmitir todos los paquetes que quedan por enviar. De hecho, que todos los paquetes sean transmitidos dentro del retardo máximo especificado en el TSPEC depende del ancho de banda síncrono asignado y el ancho de banda asíncrono disponible.

El ancho de banda síncrono, tanto para los flujos de tráfico CBR como para los flujos VBR, se calcula como la mínima capacidad que garantiza la tasa de datos promedio ρ_i , esto es [5]:

$$H_i = tx(P)I(i) + \left\lceil \frac{\rho_i * TTRT}{L_i} \right\rceil tx(L_i, R_i) \quad (20)$$

Donde $tx(P)$ es el tiempo necesario para transmitir una trama QoS CF-Poll e $I(i)$ es una función que toma el valor uno cuando el índice i es un flujo ascendente, y el valor cero si es descendente. Además, para el cálculo del ancho de banda síncrono se usa la tasa física de transmisión (R_i) indicada en el TSPEC. Utilizando esta tasa, se garantiza la tasa promedio ρ_i en el peor de los casos [5].

Una vez que se han desarrollado los conceptos de ancho de banda síncrono y asíncrono, se continúa con la explicación del cálculo del tiempo de transmisión, que dependerá del tipo de tráfico:

- Tráfico VBR:
Este tráfico se caracteriza por ser servido según su tasa promedio. Además, su tasa de transmisión varía debido al uso de paquetes de tamaño variable o a la modificación de su intervalo de servicio. Por tanto, además de un ancho de banda síncrono es necesario asignarle un ancho de banda asíncrono que permita soportar las ráfagas características de este tipo de tráfico. Para ello, cuando el servidor visita un flujo de tráfico i con tráfico VBR, calcula el tiempo de transmisión sumando el ancho de banda síncrono H_i (fijo para todos los ciclos) y el ancho de banda asíncrono (calculado en cada vuelta). Con el ancho de banda síncrono se garantiza una tasa suficiente para cumplir con la condición de la tasa promedio del TSPEC. Añadir un ancho de banda asíncrono al tráfico VBR permite amortiguar el aumento del retardo provocado por las ráfagas.
- Tráfico CBR:
Se le asigna únicamente ancho de banda síncrono.
- Tráfico con contención:
Sólo se le asigna ancho de banda asíncrono. Por tanto, puede darse el caso de que no se le dé ninguna oportunidad de transmisión en un ciclo completo (cuando el resto de tipos de tráfico ocupan todo el ancho de

banda disponible). Sin embargo, puede usar cualquier ancho de banda que no se haya reservado o que esté sin usar.

A continuación se expone el pseudo-código del planificador en el QAP (véase la figura 4.2), encargado de repartir las oportunidades de transmisión entre los distintos flujos de tráfico y calcular la duración de estas TXOPs:

actualizar (nodo i)

```
1  i.TRT -= tiempo_actual - i.tiempo_ultimo
2  i.tiempo_ultimo = tiempo_actual
3  if ( i.TRT < 0 )
4      y = 0
5      i.TRT = i.TRT - floor(i.TRT / TTRT) * TTRT
6  else
7      y = i.TRT
8      i.TRT = TTRT
9  return y
```

servir (nodo i)

```
10  if ( i.tipo == CBR )
11      x = i.H
12  else
13      y = actualizar (i)
14      if ( i.tipo == VBR )
15          x = min (i.H + y, TTRT)
16  else
17      dormir (y)
18      return
19  if ( i.tipo == CBR || i.tipo == VBR )
20      if ( i.direccion == DOWNLINK )
21          transmitir (i, x)
22      else
23          poll (i, x)
24  return
```



```
1  i.TRT -= tiempo_actual - i.tiempo_ultimo
```

Se actualiza el valor del temporizador *TRT* para el nodo *i*, calculando el tiempo transcurrido desde el último servicio.

```
2  i.tiempo_ultimo = tiempo_ultimo
```

Se actualiza el último tiempo de servicio al nodo a partir del tiempo actual.

```
3  if ( i.TRT < 0 )
4      y = 0
5      i.TRT = i.TRT - floor(i.TRT / TTRT) * TTRT
6  else
7      y = i.TRT
8      i.TRT = TTRT
```

En este trozo de código se distinguen dos opciones:

- Contador $TRT < 0$
Si el contador *TRT* es menor que cero, ha pasado más tiempo del permitido entre dos turnos consecutivos para el nodo *i*. Por tanto, no queda capacidad excedente por lo que no se le puede asignar ancho de banda asíncrono, $y=0$. El nuevo valor de *TRT* es el que da la instrucción 5 (si ha pasado algo más de un ciclo, sería el tiempo restante del siguiente ciclo).
- Contador $TRT \geq 0$
Si el contador *TRT* es mayor o igual que cero, se asigna al nodo un ancho de banda asíncrono igual al de *TRT*. En este caso, desde el último turno concedido al nodo, no ha pasado un tiempo mayor que el valor de *TTRT* (hay ancho de banda disponible), por lo que sí se transmite usando ancho de banda asíncrono.

```
9  return y
```

La función devuelve el valor actualizado de *y*.

La función *servir (nodo i)* es la encargada de calcular la duración de la TXOP proporcionada por el coordinador híbrido a cada QSTA para que pueda transmitir sus tramas. Para ello, distingue entre tráfico CBR, VBR y EDCA/DCF:

- Si el tráfico es CBR, la duración de la TXOP tendrá un valor igual a *H*. Si es un flujo de tráfico de enlace descendente, se transmitirá sin necesidad de transmitir un mensaje de *polling*. Si es un TS de enlace ascendente, el QAP tendrá que enviar primero un mensaje de *polling* indicándole a la estación cuánto tiempo tiene disponible para transmitir.
- Si el tráfico es VBR, primero se actualiza el valor *y* del nodo. El valor de la TXOP será el valor obtenido por la instrucción $x = \min(i.H + y, TTRT)$. A partir de aquí se trata igual que al tráfico CBR. Es decir, si es un TS de enlace descendente, se transmitirá sin enviar un mensaje de *polling*. Si es un TS de enlace ascendente, habrá que enviar un mensaje de *polling* indicándole a la estación cuánto tiempo tiene disponible para transmitir.

- Si el tráfico es EDCA/DCF, la función *dormir* reservará para este tipo de tráfico el medio durante un tiempo γ .

Este algoritmo tiene una baja complejidad computacional. El procedimiento *servir* necesita un número constante de operaciones por cada flujo, y la función *servir* se invoca al final de cada TXOP (o período de contención). Sin embargo, la carga computacional de WTTP [5] es menor que la de otros algoritmos que tratan eficientemente el tráfico VBR.

En los siguientes apartados del proyecto se analizará el rendimiento de WTTP mediante simulación.

5. Network Simulator 2

En los puntos anteriores se han visto los conceptos teóricos que comprende el proyecto: estándares IEEE 802.11 e IEEE 802.11e, algoritmos de planificación de paquetes en HCCA y descripción del mecanismo seleccionado WTTT.

A continuación, WTTT será analizado utilizando la herramienta *Network Simulator 2* (*NS2*). Para ello se usará el módulo *ns2hcca* [12] que permite utilizar la función HCCA en *NS2*. Tras su instalación, se implementará el mecanismo WTTT modificando el código del algoritmo de referencia para IEEE 802.11e.

Posteriormente, se configurarán los valores del TSPEC y se modificará el archivo de simulación TCL, para realizar las simulaciones necesarias.

5.1 Características generales

En el desarrollo del proyecto se ha utilizado *Network Simulator 2*, concretamente la versión 2.29. *Network Simulator* es un simulador de redes de eventos discretos y orientado a objetos, desarrollado por la Universidad de Berkeley en California y escrito en C++ y OTcl. NS2 se puede utilizar para simular tanto redes de área local como redes de área amplia. Además, puede simular tráfico TCP o UDP, y aplicaciones como FTP (Protocolo de Transferencia de archivos) y Telnet. También incluye mecanismos de gestión de colas, y algoritmos de enrutado. Implementa algunos de los protocolos de la capa MAC para la simulación de redes LAN o WLAN.

NS2 está escrito en C++ y OTcl, con un intérprete OTcl que interactúa con el usuario. El simulador se apoya en una jerarquía de clases en C++ (también llamada jerarquía compilada), y otra jerarquía de clases OTcl similar a la anterior (llamada jerarquía interpretada), estando ambas jerarquías relacionadas entre sí. Desde la perspectiva de un usuario, hay una correspondencia entre una clase en la jerarquía interpretada y una en la compilada (véase la figura 5.1). La raíz de toda esta jerarquía se encuentra en la clase *TclObject*. Los usuarios crean nuevos objetos para la simulación a través del intérprete; estos objetos son instanciados dentro de OTcl, y son generados simétricamente por su objeto correspondiente en la jerarquía compilada. La jerarquía de clase interpretada se establece automáticamente a través de métodos definidos en la clase *TclClass*. Los objetos instanciados simétricamente en clases compiladas lo hacen a través de métodos definidos en la clase *TclObject*.

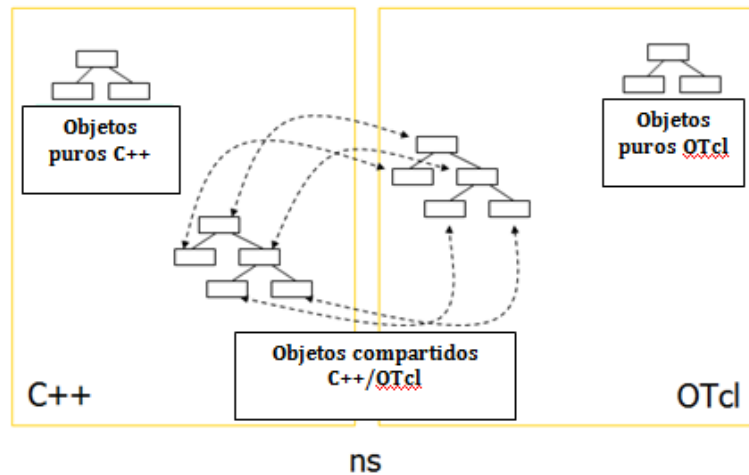


FIG5.1: DUALIDAD ENTRE CLASES

La razón por la que NS2 usa dos lenguajes diferentes se debe a la existencia de dos necesidades distintas. Por un lado, los detalles de simulación requieren un lenguaje de programación que pueda manipular con efectividad bytes, cabeceras de paquetes e implementar algoritmos que puedan tratar con efectividad grandes cantidades de datos. Para poder realizar esta tarea, la velocidad de ejecución es importante, pero no es tan importante el tiempo empleado en configuración. Por otro lado, una gran parte de la simulación en NS2 implica variar algunos parámetros de configuración para poder analizar distintos escenarios. En este caso, el tiempo empleado por el usuario es más importante. Sin embargo, ya que la ejecución se produce una sola vez (al comienzo de la simulación), el tiempo de ejecución de esta parte es menos importante. C++ es un lenguaje de programación rápido en ejecución, haciéndolo aconsejable para detalles de implementación de protocolos. OTcl es más lento en ejecución pero se puede cambiar rápidamente (e interactivamente), haciéndolo ideal para la configuración de la simulación.

Cuando una simulación termina, NS2 crea uno o más archivos de texto de salida que contienen información detallada de los datos de simulación. Estos archivos pueden ser usados para analizar la simulación o como entrada de un simulador gráfico incorporado en NS2 llamado Animador de Red (NAM). NAM tiene una interfaz de usuario en la que se puede presentar gráficamente información como la topología de la red (los nodos y sus enlaces) o el envío y descarte de paquetes (véase la figura 5.2).

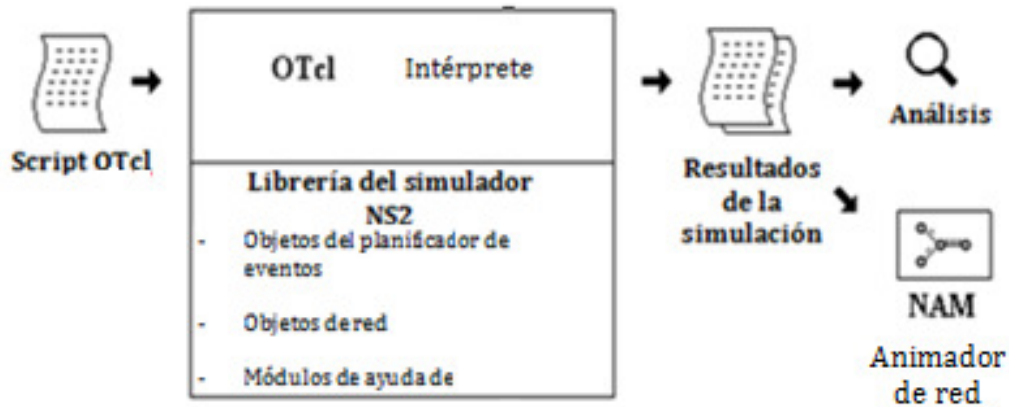


FIG5.2: VISTA SIMPLIFICADA DE LOS ELEMENTOS DE NS2

La figura 5.3 muestra la estructura de directorios que utiliza *Network Simulator 2*.

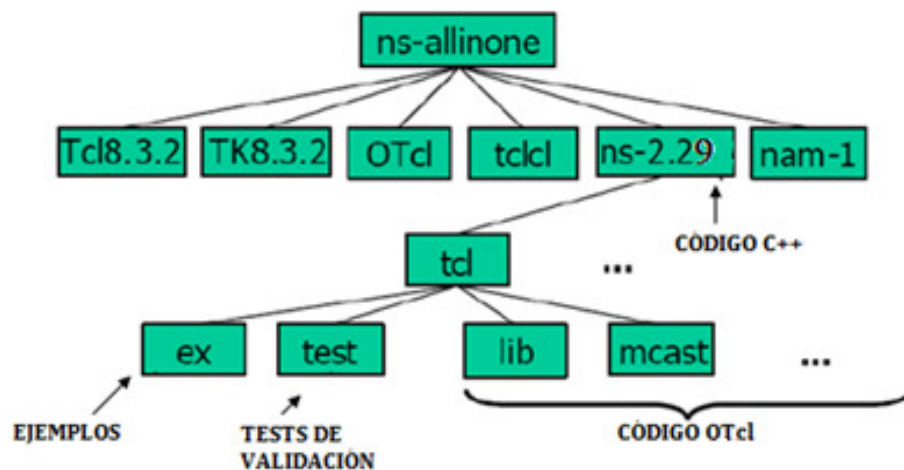


FIG5.3: ESTRUCTURA DE DIRECTORIOS DE NS2

Entre los subdirectorios de *ns-allinone-2.29*, ns-2.29 es el directorio que contiene el código del simulador (archivos C++ u OTcl), y varios ejemplos de *scripts* OTcl para validar distintas funcionalidades. Dentro de este directorio, los ficheros OTcl se encuentran en la carpeta *tcl*.

5.1.1 Ejemplo de *script* de simulación

Para este ejemplo se considera la siguiente topología de red.

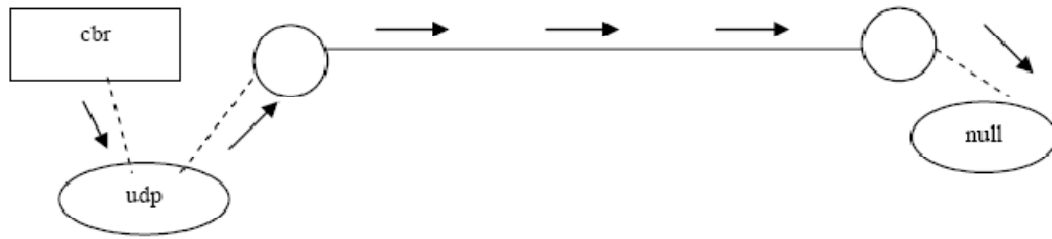


FIG5.4: ESCENARIO PARA UN SCRIPT SIMPLE DE SIMULACIÓN

En este escenario hay dos nodos: $n0$ y $n1$. El enlace *full-duplex* entre los nodos $n0$ y $n1$ tiene 2Mbps de ancho de banda y 10 ms de latencia. Cada extremo del enlace se modela como una cola de tipo *DropTail* con una capacidad máxima de 10 paquetes. El nodo $n0$ tiene un agente *UDP* para la transmisión, mientras que el nodo $n1$ cuenta con un agente *NULL* para la recepción. El agente *NULL* solamente recibe los paquetes enviados, no realizar ninguna operación sobre ellos. Además, se conecta una aplicación que genera tráfico *CBR* al agente UDP, que produce paquetes de 500 bytes cada segundo. La aplicación CBR se configura para que comience a generar tráfico a los 0.5 segundos y que termine a los 4.5 segundos.

El código del *script* OTcl quedaría de la siguiente forma:

```
# Crea el objeto simulador
set ns [new Simulator]

# Abre el archive de traza NAM
set nf [open out.nam w]
$ns namtrace-all $nf

# Procedimiento de finalización del tcl
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}

# Crea dos nodos
set n0 [$ns node]
set n1 [$ns node]

# Crea el enlace entre nodos
$ns duplex-link $n0 $n1 1Mb 10ms DropTail

# Configurar el agente UDP y adjuntarlo al nodo n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
```



```
$ns connect $udp0 $null0

# Configurar CBR sobre el agente UDP
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 1.0
$cbr0 attach-agent $udp0

# Planificar eventos para el agente CBR
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"

# Llamada al procedimiento de finalización después de 5 segundos
#de simulación
$ns at 5.0 "finish"

# Ejecución de la simulación
$ns run
```

A continuación se va a comentar el *script* desarrollado anteriormente:

- **set ns [new Simulator]**: instancia un nuevo objeto simulador, y lo asigna a una variable *ns*. Esta línea crea un nuevo planificador de eventos para la simulación, inicializa el formato de los paquetes y selecciona el formato de la dirección por defecto.
- **\$ns namtrace-all nf**: indica al simulador que grabe la información de simulación en una traza adecuada para poder ser tratada por NAM. En el archivo *nf* se almacena toda la información que usará posteriormente NAM para realizar sus representaciones gráficas de los resultados.
- **proc finish {}**: cierra el archivo de traza y inicia el simulador NAM.
- **set n0 [\$ns node]**: crea un nodo. Un nodo en NS2 está formado por una dirección y un puerto.
- **\$ns duplex-link node1 node2 bandwidth delay queue-type**: crea dos enlaces simplex de ancho de banda *bandwidth* y retardo *delay*, y conecta los dos nodos especificados.
- **set udp [new Agent/UDP]**: crea un agente *udp*. De esta forma los usuarios pueden crear cualquier tipo de fuente de tráfico.
- **\$ns attach-agent node agent**: adjunta un objeto agente a un objeto nodo
- **\$ns connect agent1 agent2**: conecta los dos agentes especificados. Una vez que los dos agentes se pueden comunicar, el siguiente paso es establecer una conexión de red lógica entre ellos. Esta línea establece una conexión lógica de red configurando dirección y puerto destino en cada uno de los nodos.
- **\$ns at 4.5 "\$cbr start"**: indica al agente CBR cuando comienza el envío de datos.

- `$ns at 0.5 "$cbr stop"`: indica al agente CBR cuando terminar el envío de datos.
- `$ns at 5.0 "finish"`: indica al objeto simulador que tiene que ejecutar el procedimiento *finish* al transcurrir 5 segundos de simulación.
- `$ns run`: comienza la simulación.

5.1.2 WLAN en NS2

Una vez vistas las características generales de NS2, se va a pasar a comentar la parte principal del simulador usada para la realización de este proyecto. En este apartado, se va a describir la implementación de las redes WLAN en NS2. Para este fin, se tendrán en cuenta conceptos teóricos comentados en apartados anteriores sobre redes WLAN.

Inicialmente, NS-2 fue ideado para redes cableadas. Sin embargo, el grupo Monarch de la CMU (Universidad Carnegie Mellon) desarrolló una ampliación para el análisis de redes inalámbricas donde se incluyen tanto las redes *ad hoc* como las redes de infraestructura.

Como se puede observar en la jerarquía de clases de la figura, IEEE 802.11 se implementa como una subclase de la capa MAC de NS2.

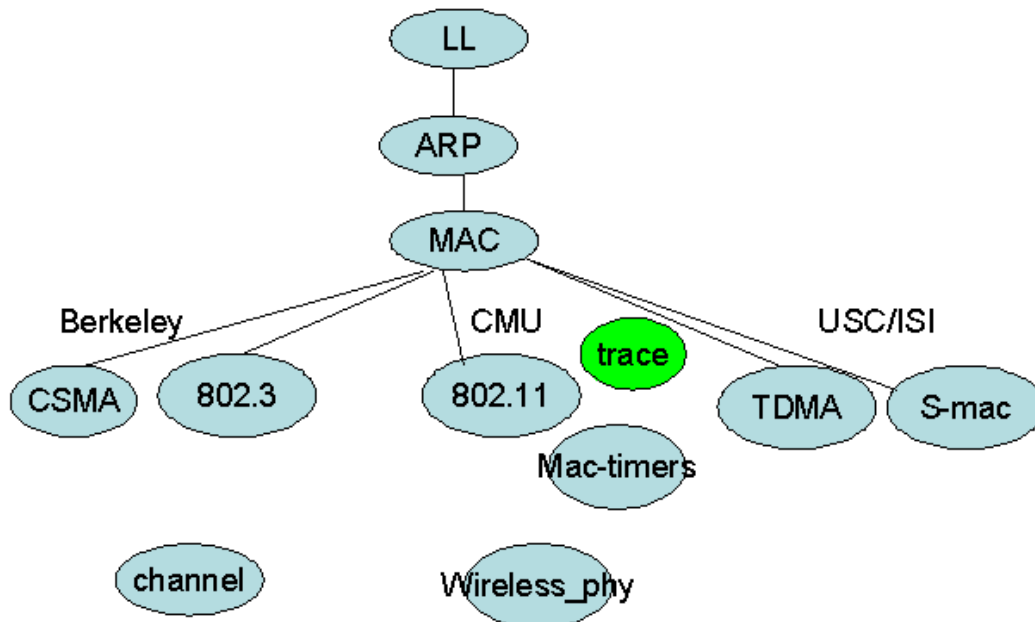


FIG5.5: CAPA MAC EN NS2

Sin embargo el modelo desarrollado para redes WLAN es muy básico y en numerosas ocasiones es necesario incluir módulos que añadan otras funcionalidades al código

original. A continuación se enumeran algunas debilidades que presenta esta implementación de las redes WLAN.

El modelo de canal desarrollado en el NS2 es tan sencillo que no tiene un comportamiento real. Un nodo móvil puede irse alejando del AP y recibiendo toda la información que éste le envíe como si no se hubiera desplazado, hasta que llega un momento en el que pasa de recibir todas las tramas a no recibir ninguna. En la realidad, este comportamiento es gradual. Conforme se va alejando un nodo móvil del AP empieza a perder tramas progresivamente hasta que deja de recibirlas.

No incluye ninguna de las extensiones de IEEE 802.11 creadas para resolver los objetivos expuestos en el apartado 2 de este proyecto. En el caso de IEEE 802.11e, es necesario incluir un módulo que posibilite el uso de la función HCCA para la provisión de QoS.

El objetivo del siguiente apartado es comentar los módulos añadidos a NS2, que permiten implementar funcionalidades el mecanismo HCCA y la generación de tráfico VoIP. Todos ellos pertenecen al Grupo de Redes de Ordenadores de la Universidad de Pisa [12][14]. Estos módulos modifican la versión original de NS2.

5.2 Módulos añadidos

A pesar de todas las funcionalidades que tiene disponibles, NS2 no incluye el mecanismo HCCA que utilizamos para poder garantizar requisitos de QoS a los diferentes flujos de tráfico admitidos. Para poder hacer uso de esta función, hay que añadir un módulo que modifica la implementación original de NS2. Este parche se encuentra disponible en [12], donde además se describen los pasos que hay que seguir para poder incluirlo en NS2.

Seguidamente, se va a realizar una breve descripción de las clases principales que permiten la implementación de HCCA en NS2 y que se añaden al incluir este módulo [12]. Estas son las clases que se van a modificar para poder implementar el algoritmo WTTT. Se exponen en la figura 5.1.

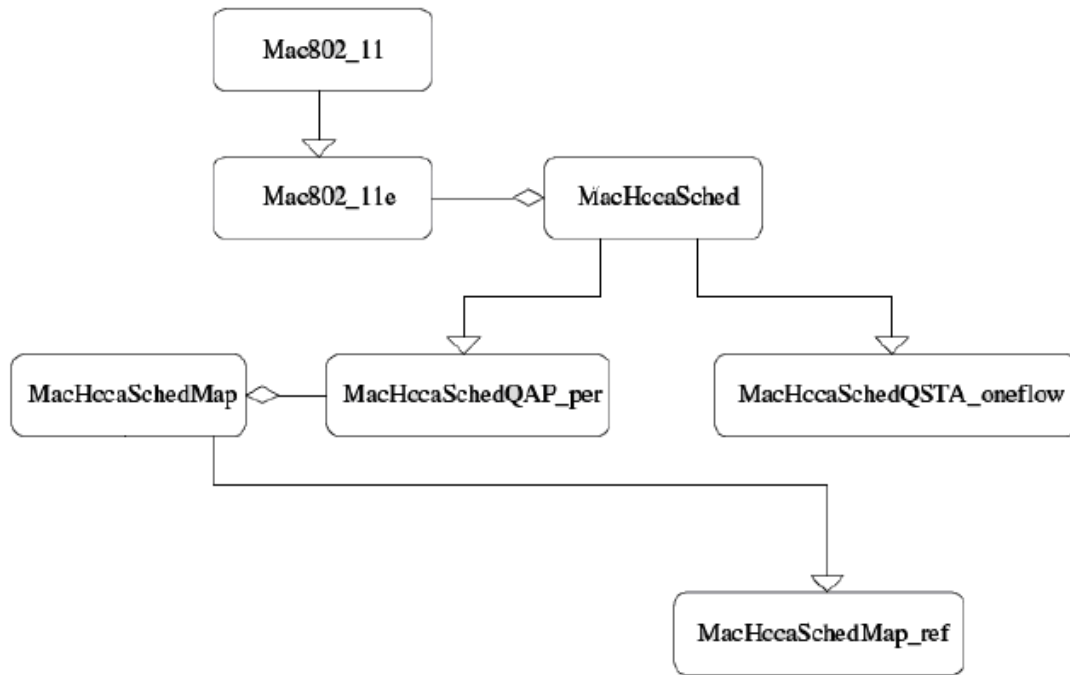


FIG5.6: DIAGRAMA DE CLASES HCCA

- *Mac802_11*: en esta clase se implementan las funcionalidades de la capa MAC de IEEE 802.11. Para poder utilizar el estándar IEEE 802.11e hay que añadir extensiones.
- *Mac802_11e*: es la clase que se añade para implementar la función HCCA, usada por el algoritmo WTTT.
- *MacHccaSched*: es una interfaz que contiene todas las funciones comunes que cualquier algoritmo de planificación debe tener para poder usar HCCA. De esta forma se aísla la capa MAC del algoritmo implementado.
- *MacHccaSchedQSTA_oneflow*: esta clase representa el planificador usado por la QSTA. Se le llama *oneflow* porque gestiona como máximo un flujo de tráfico por cada dirección. Esto permite poder centrarse en el rendimiento del planificador del QAP.
- *MacHccaSchedQAP_periodic*: esta clase se encarga de asignar a los flujos de tráfico las diferentes oportunidades de transmisión. Implementa todas las funcionalidades necesarias para servir el tráfico en el tiempo actual usando el planificador de referencia del estándar IEEE 802.11e.
- *MacHccaSchedMap*: esta clase proporciona a la clase *MacHccaSchedQAP_periodic* una estructura de datos que representa a los flujos de tráfico.
- *MacHccaSchedMap*: esta clase permite calcular las TXOPs, su duración y su tiempo de inicio. Además, en esta clase se encuentra el control de admisión del mecanismo HCCA.
- *MacHccaSchedPar*: se usa para mapear parámetros necesarios para otras clases.

NS2 versión 2.29 no incluye la funcionalidad para generar tráfico VoIP. Para ello, se debe añadir un nuevo módulo [14]. Su parche está disponible para otra versión de NS2, por lo que su inclusión se hizo manualmente para la versión 2.29, analizando línea por línea el código del parche e integrándolas en las librerías y archivos de esta versión.

A continuación se muestra el esquema del generador de tráfico VoIP para NS2:

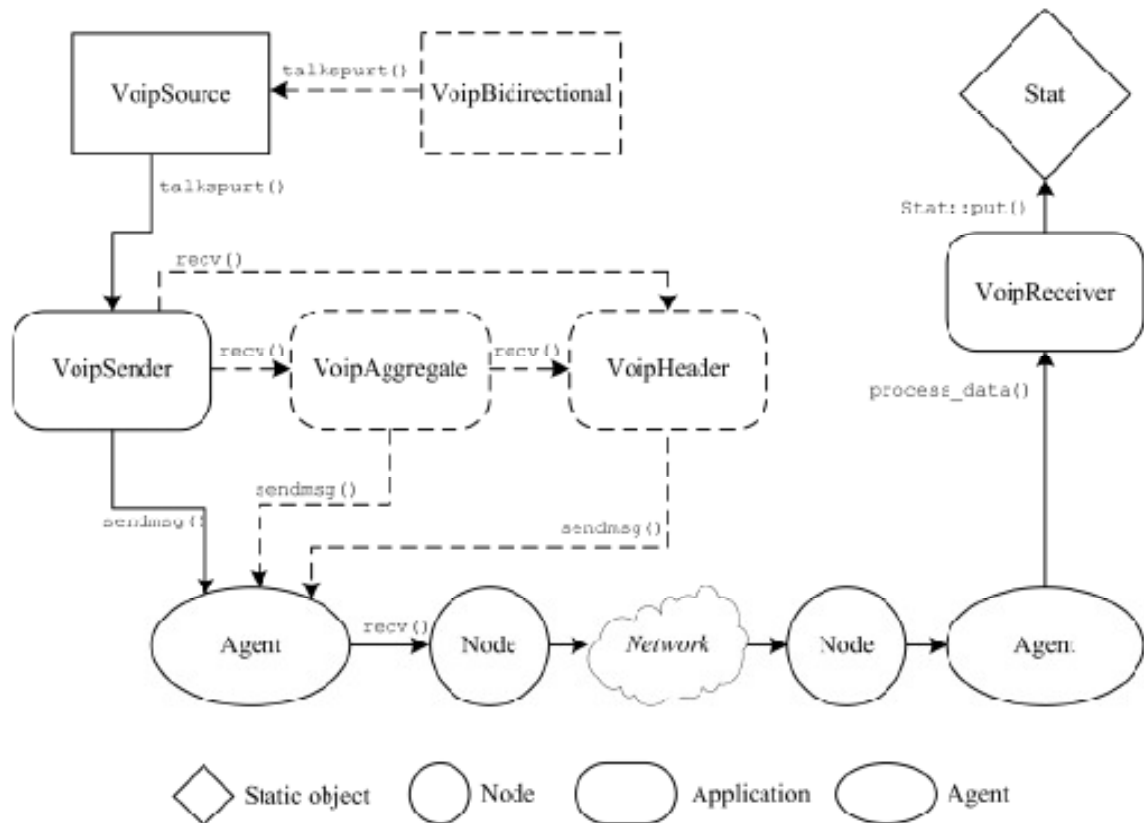


FIG5.7: ESQUEMA DE VOIP PARA NS2

La llamada telefónica es modelada mediante periodos alternativos de sonido y silencio que son generados por el objeto *VoipSource*. En el comienzo de cada intervalo de sonido el objeto *VoipSource* llama a la función *talkspurt()*, indicando la duración de los dos tipos de intervalos. Cuando esto ocurre, el objeto *VoipSender* comienza a generar una secuencia de elementos *VoipFrame*, cuyo tamaño y tasa de generación depende del códec simulado (G.711, G729, etc.).

El objeto *VoipAggregate* añade la cabecera (RTP/UDP/IP) al tráfico generado por el objeto *VoipSource*. Posteriormente el agente origen convierte este objeto generado a un paquete de NS2.

Los paquetes que llegan al destino son recibidos por el objeto *VoipReceiver*, para después procesar los datos y obtener estadísticas con la clase *Stat*.

Una vez vistos los parches incluidos en el código de NS2, en el siguiente punto se explican los cambios introducidos por el propio algoritmo WTTTP.

5.3 Modificación de NS2: implementación de WTTT

Han sido numerosas las modificaciones realizadas al código del algoritmo de referencia de IEEE 802.11e [12]. Los cambios más significativos se encuentran en el archivo *mac-hccasched_map_ref.cc*, concretamente en el algoritmo de control de admisión y en la función *next_txop (void)*. La primera función se encarga de permitir que un flujo pueda ser admitido o no por el algoritmo para su planificación comprobando la fórmula 17. La segunda función modificada se encarga de comprobar cuál es el siguiente nodo que tiene paquetes para transmitir o que, según el algoritmo WTTT, debe recibir una oportunidad para transmitir. Para ello utiliza una cola en la que están todos los nodos existentes en la red y comprueba cuál de estos nodos quiere transmitir o no utilizando la función *quiere_transmitir (void)*. Una vez comprobado y seleccionado un nodo que quiere transmitir, se ejecuta el algoritmo WTTT descrito en el apartado 4.4.

En primer lugar se diferencia entre tráfico CBR y VBR. Si la fuente genera tráfico CBR, la duración de la oportunidad de transmisión se obtendrá mediante la fórmula 20. Si se trata de tráfico VBR, se llamará a la función *update_nodo (void)*, que actualiza el valor del contador *TRT*, y calcula el valor que tendrá la oportunidad de transmisión para este caso mediante el siguiente cálculo:

$$x = \min (i.H + y, TTRT)$$

Una vez realizados estos cálculos, se actualizan los valores del comienzo de la siguiente oportunidad de transmisión, el próximo intervalo de servicio y el valor del tiempo EDCA.

Además de los cambios realizados en el archivo *mac-hccasched_map_ref.cc*, se han realizado cambios en otros archivos de la clase *Mac802_11e* (véase el apartado 5.2), principalmente para obtener parámetros del TSPEC no utilizados en el algoritmo de referencia IEEE 802.11e, como pueden ser el tipo de tráfico (VBR o CBR) y el mínimo intervalo de servicio.

A continuación se exponen los resultados obtenidos con este simulador, con las fuentes generadoras de tráfico y los archivos de generación de estadísticas. En el siguiente apartado se harán las simulaciones requeridas para comprobar el funcionamiento y rendimiento del mecanismo WTTT con varios modelos de tráfico distinto.

6. Simulaciones y resultados

El rendimiento del algoritmo WTTP ha sido analizado mediante su simulación en NS2. Primero se definen los parámetros iniciales usados en todo el desarrollo de la simulación, posteriormente se describen los distintos escenarios y se analizan los resultados obtenidos en cada uno de ellos.

Concretamente se evalúa el comportamiento de WTTP para servir flujos de tráfico VBR, con condiciones de carga diferentes. Además, se analiza la capacidad de WTTP respecto al máximo número de flujos admitidos por el control de admisión, su sensibilidad a la selección de distintos valores de *TTRT* y a las irregularidades en el intervalo de llegada de paquetes.

Por último, se comprueba el rendimiento de WTTP en un escenario mixto que involucra fuentes CBR y VBR.

6.1 Consideraciones iniciales

Fuentes de tráfico

Se consideran dos tipos de tráfico con requisitos de QoS: VoIP (mediante el módulo añadido al código original de NS2 [14]) y video bajo demanda (VoD). Además, se utiliza el mecanismo DCF para transmitir el tráfico que no posee requisitos específicos de QoS.

El flujo de tráfico de VoIP se simula como una fuente *ON/OFF*: durante el periodo *ON* el tráfico es CBR, con parámetros que dependen del esquema de codificación. Durante el periodo *OFF* no se generan paquetes. El esquema de codificación empleado es G.711, que genera tramas de voz de 80 bytes cada 10 milisegundos. Se han utilizado fuentes que empaquetan dos tramas de voz en un paquete RTP, siendo su carga útil de 160 bytes y su periodicidad de 20 milisegundos.

Para simular la fuente de video bajo demanda, se utilizará una videoconferencia (VC) que genera tráfico usando un archivo de traza MPEG4 (LectureHQ-Reisslein [13]). Esta traza crea un flujo de tramas con tamaño variable con un intervalo de generación de tramas fijo de 33 ms (30 tramas por segundo). La tasa promedio especificada es aproximadamente 158 Kbps, aunque existen ráfagas de hasta 2.7Mbps. Hay que comentar que el archivo de traza obtenido [13], se encuentra en formato ASCII. Sin embargo, para que NS2 pueda usar este archivo, necesita que se encuentre en formato binario. Para ello se usa la aplicación Matlab expuesta en el anexo III.

Se trabajará bajo la suposición de que cada estación tiene un solo flujo y que todos los nodos están directamente conectados con el resto. Por tanto, no consideramos el problema del nodo oculto (véase el apartado 2.3.3.1).

Parámetros de simulación

Una vez descritas las fuentes generadoras de tráfico, el siguiente paso consiste en explicar las características generales de la simulación. Se han realizado simulaciones de 400 segundos, con un tiempo de inicio aleatorio para cada una de las fuentes, de forma que la generación de tráfico sea independiente del resto de fuentes.

En la siguiente tabla se resumen los parámetros de la capa MAC y de la capa física con mayor relevancia:

Parámetro	Valor
SIFS	10 μ s
PIFS	30 μ s
DIFS	50 μ s
<i>Slot Time</i>	20 μ s
Tasa de datos	11 Mb/S
Tasa básica	1Mb/s
CWmin	31
CWmax	1023

TABLA 6.1: PARÁMETROS DE LA CAPA MAC Y DE LA CAPA FÍSICA

Generación de resultados

A partir de todos estos elementos descritos, el algoritmo WTTTP implementado en *NS2* genera archivos de trazas en formato ASCII. En estos archivos se registran cada uno de los eventos ocurridos (tipo de evento, tiempo cuando este evento ocurre, etc.). Estos datos pueden ser manipulados para extraer información relevante (e.g. con alguna herramienta para procesamiento de ficheros de texto como *AWK*) y construir gráficos (con herramientas como *GNU PLOT* o *MS Excel*). El fichero *awk* desarrollado se presenta en el Anexo II. Este fichero genera, entre otras, estadísticas como el retardo medio, tasa de transferencia y el percentil 99 del retardo. A continuación se expone un cuadro resumen con el formato de la información generada por *NS2* (para el caso de redes inalámbricas, ya que para redes cableadas es distinto):

Evento	Abreviatura	Flag	Valor
Evento inalámbrico	s: enviado r: recibido d: descartado f: transmitido	- t	Tiempo
		- Ni	Identificación del nodo
		- Nx	Coordinada X del nodo
		- Ny	Coordinada Y del nodo
		- Nz	Coordinada Z del nodo
		- Ne	Nivel de energía del nodo
		- NI	Nivel de traza de red
		- Nw	Razón del descarte
		- Hs	Identificación de la fuente
		- Hd	Identificación del destino
		- Ma	Duración
		- Ms	Dirección Ethernet de la fuente
		- Md	Dirección de Ethernet del destino
		- Mt	Tipo de Ethernet
		- P	Tipo de paquete
		- Pn	Tipo de paquete

TABLA 6.2: FORMATO DE LAS TRAZAS DE SALIDA DE NS2

6.2. Escenarios y resultados

A continuación se describen los diferentes escenarios simulados y los resultados obtenidos en cada uno de ellos. Hay que indicar que estos escenarios son una reproducción de los usados en [5]. A partir de estos escenarios se va a comprobar que el funcionamiento del algoritmo implementado.

6.2.1. Escenario 1

En este escenario se va a analizar únicamente el rendimiento del enlace ascendente con tráfico VBR (sólo video bajo demanda, fichero de traza de vídeo *example-trace*). El retardo medio se especifica en el TSPEC del archivo *examplecompleto.tcl*, siendo su valor de 33 ms. Tanto el mínimo intervalo de servicio como el máximo intervalo de servicio tienen un valor de 16.67 ms (valor obtenido de [5]). La velocidad de tasa promedio es de 158 Kbps y la tasa de la capa física es de 1Mbps.

Con estos parámetros de configuración se analiza el rendimiento del tráfico VBR, concretamente, el retardo y el intervalo de *polling* (véase la figura 6.1 y compárese con los resultados de [5]). El retardo consiste en el tiempo transcurrido desde que el

paquete alcanza la capa MAC del origen, hasta que se comprueba que ha llegado al destino correctamente. Por otro lado, el intervalo de *polling* para el enlace ascendente se define como el tiempo transcurrido entre dos oportunidades de transmisión consecutivas asignadas al mismo flujo de tráfico.

Según la definición de ancho de banda asíncrono de la expresión 20, cada flujo de tráfico ha de transmitir al menos a una velocidad igual a su tasa promedio de bits que es mucho menor que la tasa de pico. Por tanto, como podemos observar en la gráfica 6.1, concretamente analizando el valor del percentil 99, existe una serie de paquetes con un retardo mayor que 33 milisegundos (retardo medio especificado en el TSPEC).

Tanto el percentil 99 como el retardo promedio aumentan con el número de estaciones admitidas. El percentil 99 del retardo puede verse como el retardo máximo obtenido en el caso peor. Este retardo máximo sería la suma de dos componentes. Por un lado, el intervalo de generación de mensajes de *polling*, cuyo valor máximo viene definido por el intervalo de generación de paquetes (33 milisegundos con la fuente de tráfico utilizada). Esta componente tendría su caso peor cuando se genera un paquete casi inmediatamente después de enviar el último paquete en la cola, por lo que su retardo por este motivo coincidiría aproximadamente con el intervalo de generación de *polling*, i.e. 33 milisegundos en este escenario. La segunda componente es la duración máxima de un ciclo WTTTP a través de la lista de nodos (véase el apartado 4.2), que depende del número de estaciones admitidas. Considerando ambas componentes, se espera que este retardo máximo pueda exceder de 33 milisegundos, como se muestra en la figura 6.1.

Además de los valores de los retardos, en la figura 6.1 se presenta el valor medio del intervalo de *polling*. En el caso general, el intervalo de *polling* coincide con la duración de un ciclo completo (*TTRT*, 16.67 milisegundos en este escenario), ya que cada flujo en sentido ascendente transmite una vez por ciclo. Éste aumenta ligeramente al aumentar el número de estaciones, debido a que aumenta el tiempo en pasar el turno por la lista WTTTP [5].

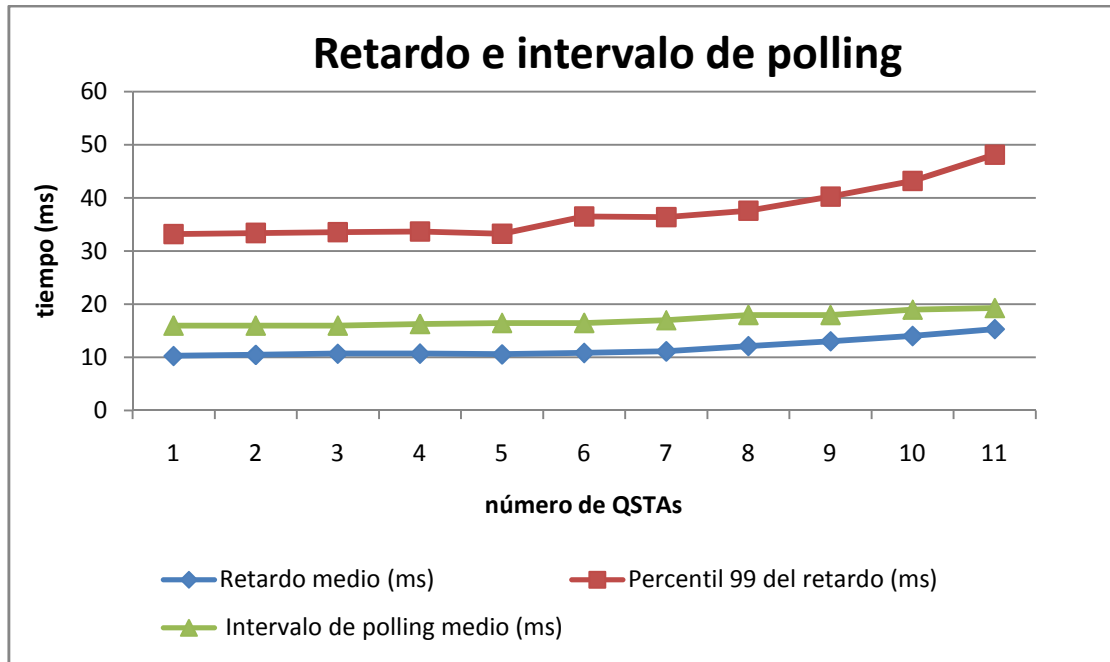


FIG6.1: RETARDO (TANTO DEL PERCENTIL 99 COMO DEL PROMEDIO) E INTERVALO DE POLLING DEL ENLACE ASCENDENTE CON UN FLUJO VBR

La gráfica 6.2 se ha obtenido usando el *script analyze.pl*, que calcula el *throughput* para cada uno de los flujos. Como se puede comprobar el *throughput* está garantizado, a pesar de que aumenten el número de estaciones (gracias al control de admisión). La grafica 6.2 verifica en este escenario que WTTP es un algoritmo que garantiza la tasa de bit, como se comentó en el apartado 4.

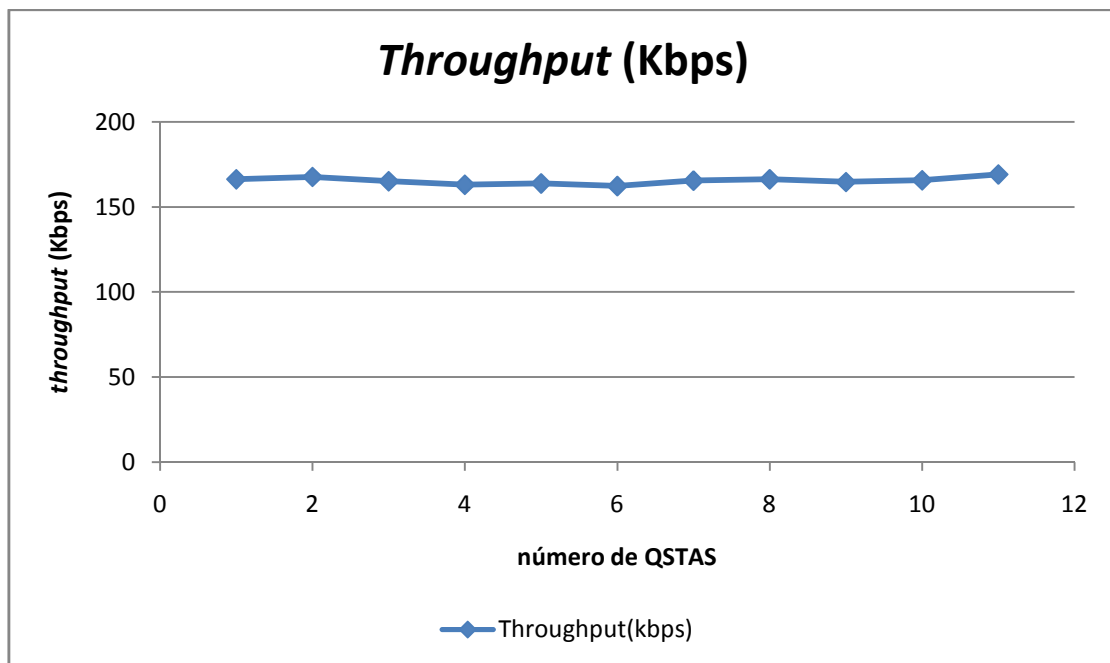


FIG6.2: THROUGHPUT DEL ENLACE ASCENDENTE CON TRÁFICO VBR

En la figura 6.3 se muestra el tiempo dedicado a la transmisión de tráfico mediante el mecanismo EDCA por cada ciclo, es decir, cada intervalo de tiempo TTRT. Como se puede observar, éste va disminuyendo a medida que aumentan el número de estaciones admitidas, ya que, al aumentar el número de estaciones admitidas, el ancho de banda que queda disponible para EDCA, disminuye.

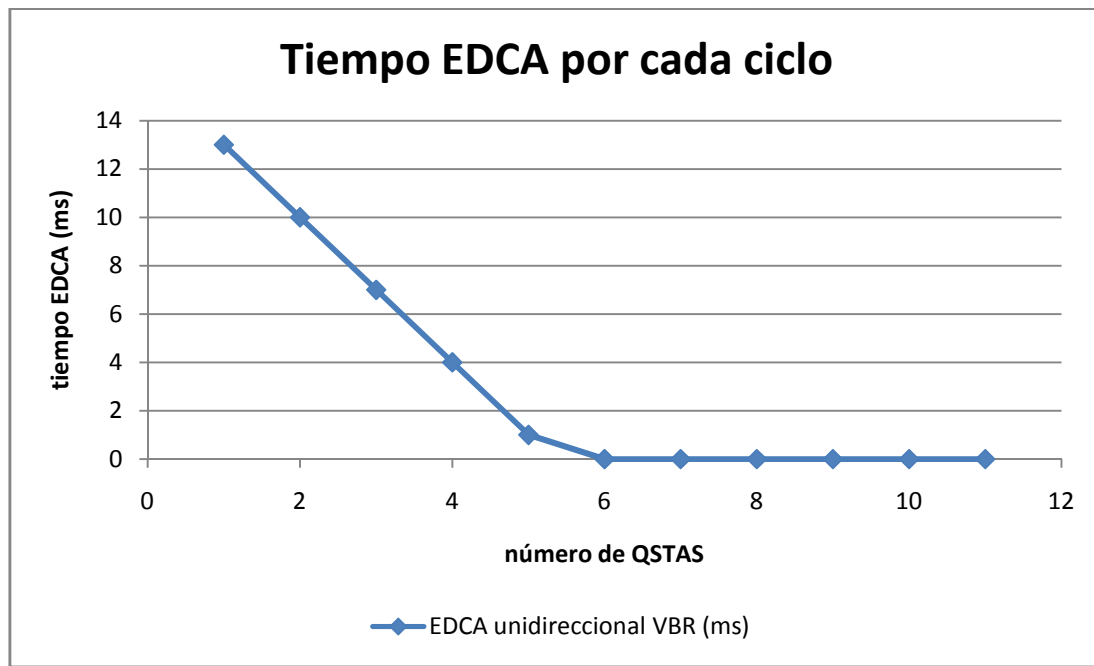


FIG6.3: TIEMPO EDCA DEL ENLACE ASCENDENTE CON TRÁFICO VBR

Para concluir el análisis del algoritmo WTTTP con el primer escenario, en la figura 6.4 se presenta la función de densidad de probabilidad acumulada (CDF) del retardo. Se analizan tres casos, 1QSTA, 11 QSTAS (máximo número de QSTAs admitidas) y 13 QSTAs (límite máximo soportado por el control de admisión de FHCF, comparado con WTTTP en [5]). Como se puede observar en la figura, los valores del retardo no superan en ninguno de los tres casos los 50 ms. Por tanto el percentil 99 es estable a pesar de que se aumente el número de estaciones.

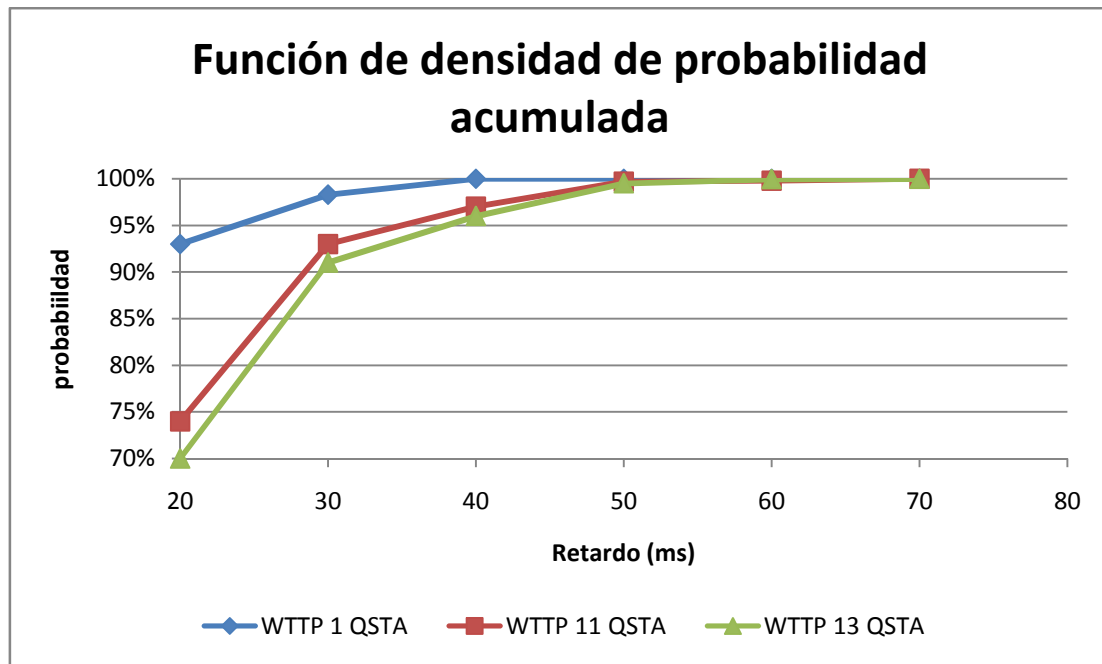


FIG6.4: CDF DEL RETARDO DEL ENLACE ASCENDENTE CON TRÁFICO VBR

6.2.2. Escenario 2

En este escenario se va a analizar la sensibilidad de WTPP respecto al valor del parámetro TTRT (compárese con los resultados obtenidos en [5]). Para ello, se va a repetir la simulación anterior con todos los parámetros del TSPEC idénticos a excepción del retardo máximo que esta vez tomará los siguientes valores:

- Retardo máximo → 33 milisegundos
- Retardo máximo → 36 milisegundos (+10%)
- Retardo máximo → 30 milisegundos (-10%)

De esta forma, y considerando la fórmula 18, se cambia el valor del TTRT de 16.67 ms (valor del TTRT en el primer escenario), y se aumenta y disminuye un 10% su valor para comprobar la sensibilidad a este parámetro. Además se aumenta el número de fuentes admitidas hasta 17 fuentes (se desactiva el control de admisión, ya que supera el número máximo de estaciones admitidas según el algoritmo WTPP), que es un 50% más que en el caso del escenario 1. Finalmente, se modifica la fuente generadora de tráfico, ya que en este caso el tiempo de llegada de las tramas tendrá una componente aleatoria y no un valor fijo a 33 ms. Para ello usamos la traza *example-trace-random* en la que los tiempos de llegada varían de forma aleatoria. Esta nueva traza simula un intervalo de tiempo aleatorio con una variación en el tiempo de llegada de los paquetes del 10% respecto al valor nominal ($33 \pm 10\%$) añadido por los sistemas de planificación de paquetes, lo que provoca que la información nominal no sea real.

En la figura 6.5 se muestra el percentil 99 del retardo. Como se puede observar en la gráfica una variación del TTRT del 10%, no implica cambios drásticos en el rendimiento del algoritmo. Se puede comprobar que el mecanismo permanece estable, a pesar de que se ha cambiado su valor.

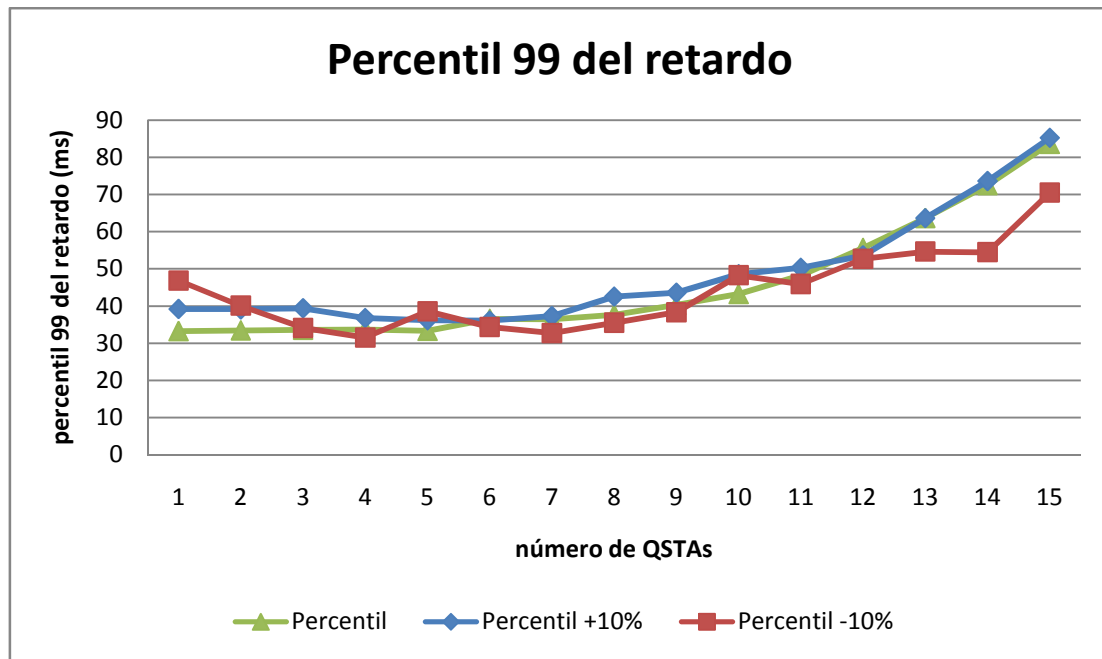


FIG6.5: PERCENTIL 99 DEL RETARDO PARA VARIOS VALORES DEL TTRT

En la figura 6.6, se muestra el retardo medio, que también demuestra la estabilidad del retardo al variar el parámetro *TTRT*. Además, se puede comprobar que al aumentar el número de estaciones aumenta el retardo debido a que queda menos ancho de banda asíncrono disponible para las ráfagas de tráfico que se reciben. Esto provoca que el retardo aumente de manera progresiva.

Por otro lado, al aumentar el valor del *TTRT*, también aumenta el retardo medio como se puede observar en la figura 6.6. Esto se debe a que el tiempo de duración de cada ciclo aumenta, y por tanto hay más tiempo disponible para poder transmitir tráfico mediante el mecanismo EDCA. Las fuentes transmiten en su turno, pero tienen que esperar más tiempo para recibir la siguiente oportunidad de transmisión. El caso contrario ocurre para el caso en el que el valor del parámetro *TTRT* sea menor.

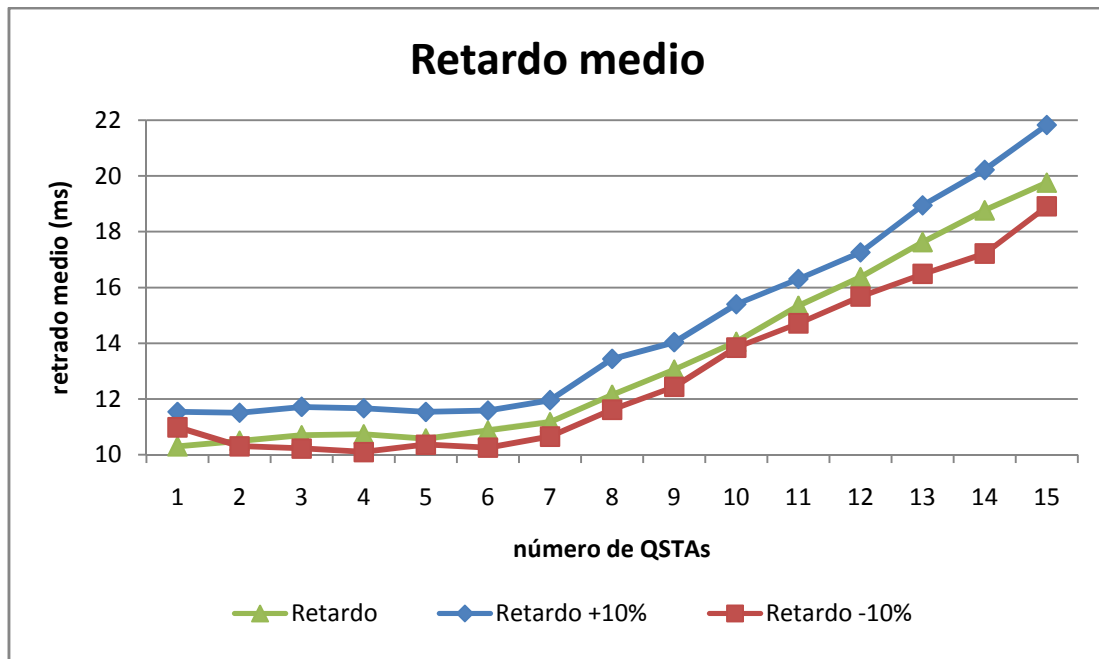


FIG6.6: RETARDO MEDIO PARA VARIOS VALORES DEL TTRT

En la figura 6.7 se muestra el *null ratio* para el enlace ascendente. Este se define como la relación entre el número de mensajes nulos (no hay datos que transmitir) enviados como respuesta a un mensaje de *polling* del QAP y el número de total de mensajes *polling* asignados a esa estación. Es una medida de la sobrecarga de mensajes de *polling* debido al esquema seguido por el algoritmo de planificación. Como se puede comprobar este algoritmo es muy eficiente, ya que el *null ratio* es muy bajo, y se muestra estable del parámetro *TTRT*.

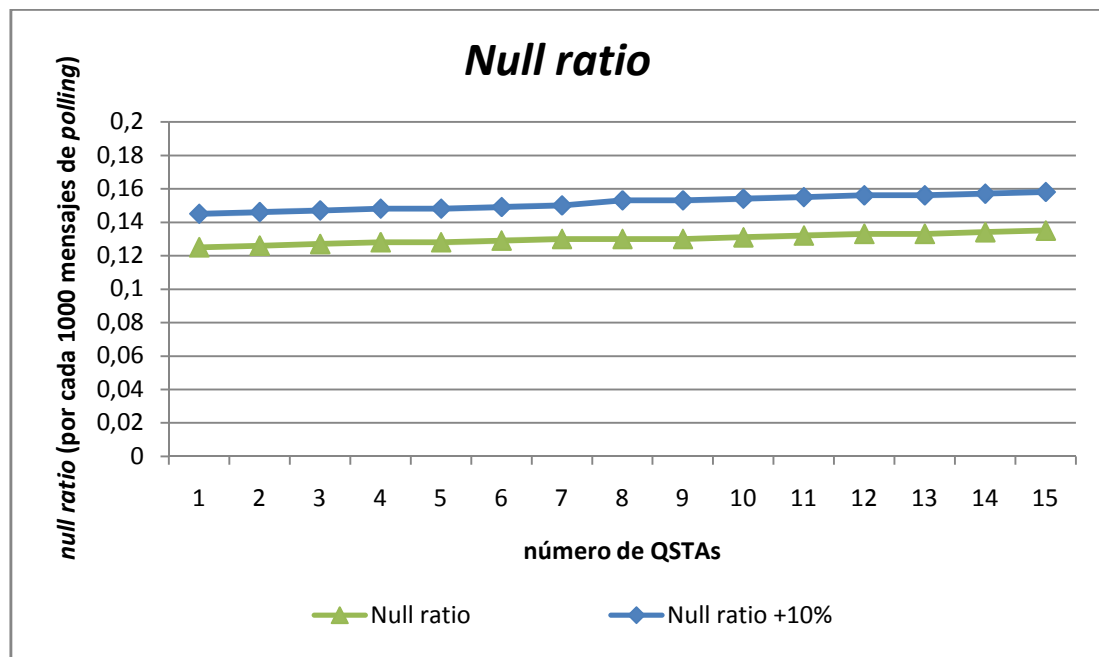


FIG6.7: NULL RATIO DEL ENLACE ASCENDENTE CON TRÁFICO VBR

6.2.3 Escenario 3

En este escenario, se va a comparar el rendimiento del algoritmo WTTP en un escenario mixto de fuentes CBR y VBR. Se simulará tráfico VoIP (variando de 1 a 6 fuentes) bidireccional. Además, se configurará una fuente de video bajo demanda también bidireccional (compárese con los resultados obtenidos en [5]).

Para ello, se usará el módulo añadido de VoIP (véase el apartado 5.2) y la traza de video *example-trace*, usada en el escenario 1.

Para el caso de la fuente de video bajo demanda, el TSPEC será configurado con los mismos parámetros que el escenario 1. Por otro lado, las fuentes de VoIP tendrán un retardo medio de 20 ms, el tamaño de la SDU de 200 bytes (las tramas están formadas por 160 bytes de datos y 40 bytes de cabeceras IP, UDP y RTP) y 80kbps de tasa promedio.

La figura 6.8 muestra el intervalo entre mensajes de *polling* para los enlaces ascendentes de los flujos de tráfico CBR Y VBR. Las curvas son casi constantes y aproximadamente iguales al intervalo de generación de paquetes de tráfico VoIP y VC. Esto contrasta con los resultados obtenidos en el escenario 1, donde el intervalo de *polling* aumenta con el número de estaciones. En este caso se debe a la presencia de tráfico CBR, que hace que la duración total de un ciclo sea menor ya que el *TTRT* es ahora de 10 ms (fórmula 18), mucho menor que el intervalo de generación de paquetes de VC que es de 33 ms, por lo que esa componente de retardo es despreciable.

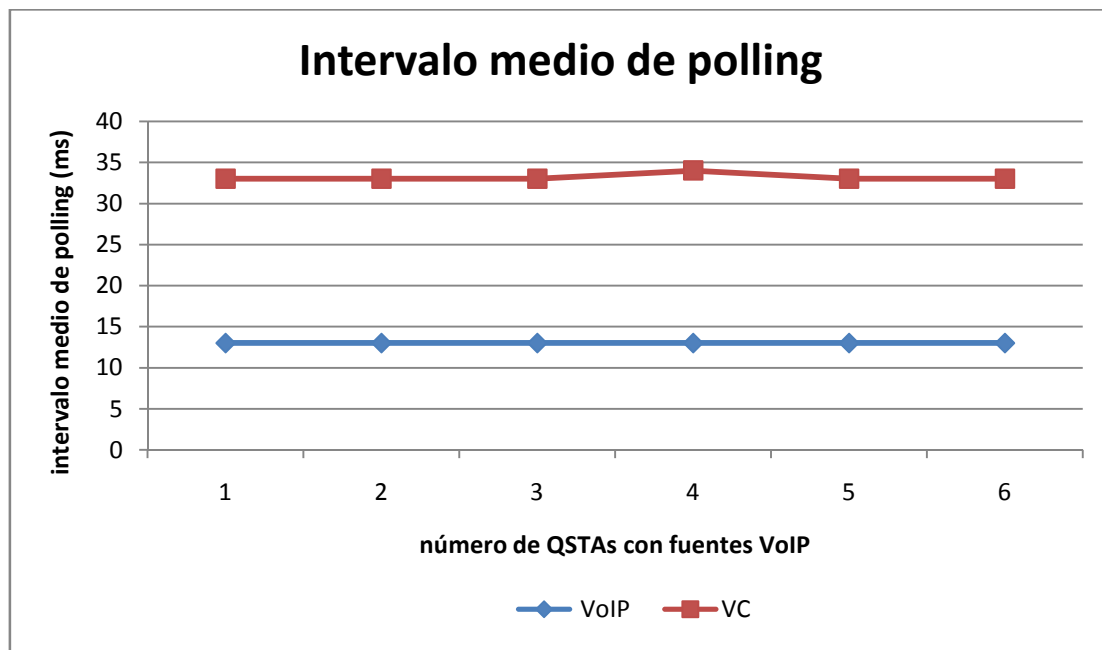


FIG6.8: INTERVALO DE POLLING PROMEDIO DE LOS FLUJOS ASCENDENTES CON TRÁFICO CBR Y VBR BIDIRECCIONAL

En la figura 6.9 se muestra el percentil 99 del retardo del tráfico VoIP y VC para los enlaces ascendente y descendente.

El enlace descendente en ambos casos, tiene un percentil 99 del retardo menor, ya que el QAP conoce el estado de la cola del flujo, y sabe en qué momento tiene paquetes para transmitir o no. Por tanto, la oportunidad de transmisión es asignada en cuanto se genera un paquete en la cola. En cuanto, al enlace ascendente para ambos casos, los turnos para transmitir vienen dados por el valor de mSI , que es de 20 ms, y por tanto, experimentan mayor retardo.

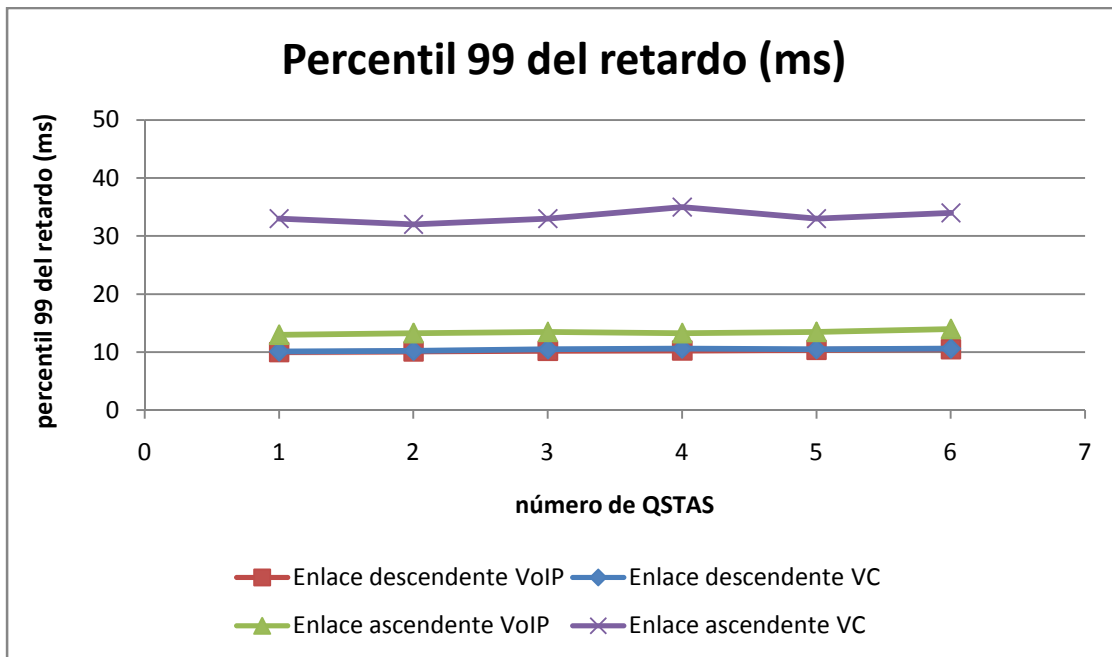


FIG6.9: PERCENTIL 99 DEL RETARDO PARA LOS ENLACES ASCENDENTES Y DESCENDENTES CON TRÁFICO CBR Y VBR BIDIRECCIONAL

En la siguiente figura, se muestra el *Throughput* medio para los flujos de tráfico CBR y VBR. Se obtienen usando el script *analyze.pl*, que calcula el *throughput* de cada uno de los flujos creados en el escenario. Como se puede comprobar el *throughput* está garantizado, a pesar de que aumenten el número de estaciones. La grafica 6.10 verifica de nuevo en este escenario que WTTP es un algoritmo que garantiza la tasa de bit como se comentó en el apartado 4.

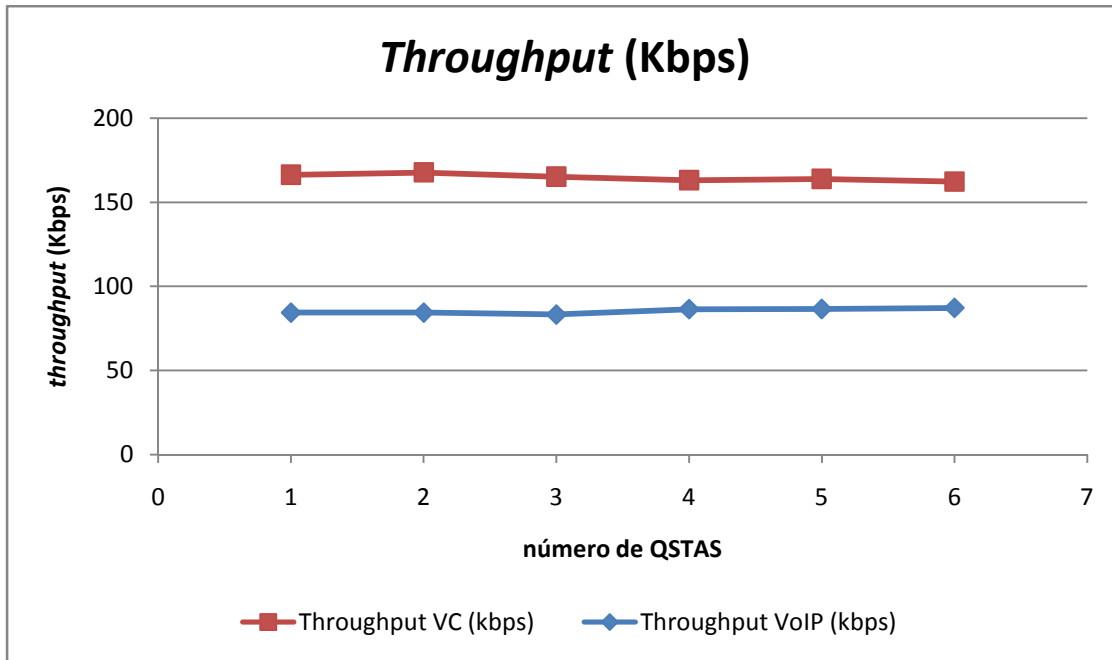


FIG6.10: THROUGHPUT MEDIO PARA LOS ENLACES ASCENDENTES Y DESCENDENTES CON TRÁFICO VBR Y CBR

En la figura 6.11 se hace una comparativa del tiempo asignado para EDCA en el escenario 1 y 2. Como se podía esperar, el tiempo EDCA disminuye en ambos casos al aumentar el número de estaciones, ya que el canal tiene que ser compartido entre un número mayor de fuentes utilizando el mecanismo HCCA.

Por otro lado se observa que en el escenario 2 el tiempo asignado para transmitir mediante el mecanismo EDCA es menor, debido a que el tráfico generado por estaciones CBR y VBR aumenta.

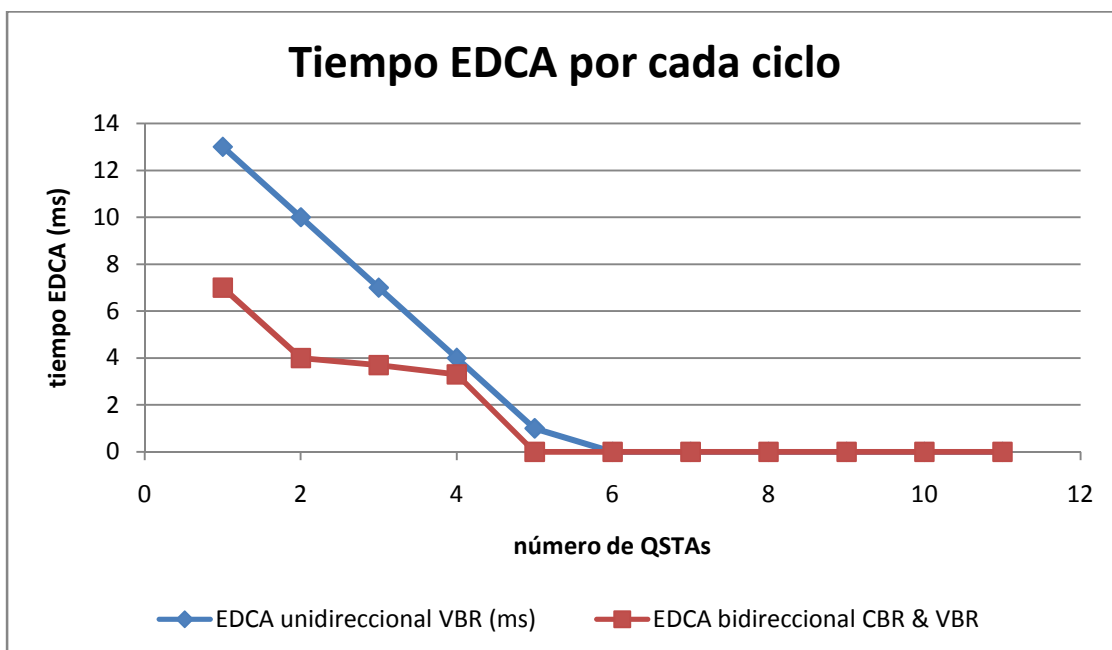


FIG6.11: TIEMPO EDCA PARA EL ESCENARIO 1 Y 3

7. Conclusiones

En este proyecto se ha realizado un estudio del estado del estándar IEEE 802.11, especialmente del estándar IEEE 802.11e. Éste ha sido diseñado para posibilitar la inclusión de servicios con requisitos de calidad de servicio en redes WLAN.

En segundo lugar, se describen los mecanismos existentes en el estándar IEEE 802.11e, EDCA y HCCA. Se ha seleccionado este último por ser el mecanismo de acceso que permite garantizar requisitos de QoS.

En tercer lugar, se hace un estudio de algunos de los principales algoritmos de planificación existentes para la función HCCA, seleccionando entre todos ellos el algoritmo WTTTP por sus características.

WTTTP es un algoritmo centralizado que puede tratar con tráfico multimedia en redes IEEE 802.11e. Además, WTTTP tiene una complejidad computacional baja [5] respecto al número de flujos de tráfico. Por otro lado es capaz de dar a los flujos admitidos, ya sea tráfico CBR o VBR, garantías de cumplimiento tanto de su tasa de bit como de su retardo máximo.

Este algoritmo proporciona al tráfico CBR un ancho de banda síncrono y constante en el tiempo. Por otro lado, al tráfico VBR se le asigna un ancho de banda extra para mantener el retardo de los paquetes bajo permitiendo la absorción de las ráfagas de llegada de paquetes características de este tipo de tráfico.

Se ha realizado su implementación en la herramienta *Network Simulator 2*, así como la evaluación de su rendimiento mediante el análisis de las simulaciones realizadas posteriormente.

Los resultados obtenidos muestran que WTTTP es capaz de servir flujos de tráfico VBR con un percentil 99 del retardo casi constante con respecto al número de flujos de tráfico admitidos.

Además, en un escenario con fuentes tanto VBR como CBR, el percentil 99 del retardo de ambos permanece estable para ambos tipos de tráfico bajo diferentes condiciones de carga.

Finalmente, se analiza la robustez de WTTTP bajo las siguientes condiciones: retardo en la llegada de paquetes aleatorio en el enlace ascendente; variaciones en el parámetro TTRT; funcionamiento del control de admisión. En el análisis se demuestra que WTTTP es capaz de comportarse bien a los efectos anteriores.

8. Referencias

- [1] J. Cowling y S.Selvakennedy. A detailed investigation of the IEEE 802.11e HCF Sample scheduler for VBR traffic, proc. IEEE ICCCN 2004, Chicago, USA, Octubre 11-13,2004.
- [2] C.Cicconetti, L.Lenzini, E. Mingozi, y G.Stea. Design and Performance Analysis of the Real-Time HCCA Scheduler for IEEE 802.11e WLANs, to appear in Computer Networks(Elsevier), Febrero 2007.
- [3] IEEE Computer Society "IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications", Aug. 1999.
- [4] A.Grilo, M.Macedo, y M.Nunes. A scheduling algorithm for QoS support in IEEE 802.11e networks, IEEE Wireless Communications, vol. 10, no.3, June 2003, pp 36-43.
- [5] C.Cicconetti,L.Lenzini,E.Mingozi,G.Stea. An Efficient Cross Layer Scheduler for Multimedia Traffic in Wireless Local Area Networks with IEEE 802.11e HCCA. University of Pisa, Italy.
- [6] Chun-Ting Chou, S. N. Shankar, K.G. Shin. Achieving per-stream QoS with distributed airtime allocation and admission control in IEEE 802.11e wireless LANs, proc. IEEE INFOCOM 2005, Miami, USA, March 1317,2005, vol.3., pp 1584-1595.
- [7] N. Ramos, D. Panigrahi, Sujit Dey. Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions, IEEE Network Magazine, vol.19, no.4, July 2005, pp.14-20.
- [8] C.Cicconetti, L.Lenzini, E. Mingozi, y G.Stea. A Scheduling Algorithm for Providing Real-time Guarantees in 802.11e WLANs.
- [9] http://www.ieeeeghn.org/wiki/index.php/Wireless_LAN_802.11_Wi-Fi
- [10] <http://www.csse.uwa.edu.au/adhocnets/802.11-1999.pdf>
- [11] Naomi Ramos, Debashis Panigrahi, y Sujit Dey. Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks. University of California, San Diego
- [12] <http://cng1.iet.unipi.it/wiki/index.php/Ns2hcca>
- [13] <http://trace.eas.asu.edu>
- [14] <http://cng1.iet.unipi.it/wiki/index.php/Ns2voip>
- [15] IEEE Standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE 802.11 std., 1999
- [16] IEEE Standard 802.11e: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), IEEE 802.11e std., 2003
- [17] <http://www.isi.edu/nsnam/ns/>

Anexo I .- script OTcl

```
#
# examplecompleto.tcl
#
# autor: José Antonio Rodríguez Salas
#
# uso: ns example.tcl [-n N] [-m M] [-v V] [-c C] [-h hcca]
# [-run R]
# donde N es el número de fuentes CBR, M el número de fuentes
# FTP, V el número de fuentes VoIP, C el número de fuentes VoD
# El parámetro hcca hay que ponerlo a 1 si se quiere usar el
# mecanismo hcca
#
#####
#                               PARÁMETROS DE CONFIGURACIÓN VOIP
#####
#
# Simulation environment
#
set opt(duration)    180.0      ;# run duration, in seconds
set opt(debug)       ""
set opt(aggregate)   2
set opt(tagrand)     "constant"
set opt(tagmean)     0.10
set opt(tagvar)      0.01
set opt(tagper)      0.00
set opt(codec)       "G.711" ;# G.711, G.723.1, G.729A, ...

#####
#                               CONFIGURACIÓN DE PARÁMETROS FTP Y CBR
#####
source header2
set val(agenttrace)  ON

#default argument options

set opt(start)       1.0        ;# traffic start time VOIP
set opt(stop)        400        ;# finaliza VOIP
set opt(n)            0          ;# number of real-time stations
(VoIP sources)
set opt(m)            0          ;# number of non-real time
stations (FTP sources)
set opt(v)            0          ;# number of VOIP stations
set opt(c)            0          ;# number of VC stations
set opt(hcca)         1          ;# HCCA is on
getopt $argc $argv
set val(n)            $opt(n)    ;# VoIP sources
set val(m)            $opt(m)    ;# FTP sources
set val(v)            $opt(v)    ;# VOIP sources
set val(c)            $opt(c)    ;# VC sources
```

```
#
# crear conexiones
#

proc create_connections {} {
    global ns opt val qsta qap defaultRNG      ;# input
    #
    # Estaciones CBR
    #
    for { set i 0 } { $i < $opt(n) } { incr i } {
        for { set j 0 } { $j < 2 } { incr j } {
            set k [expr 2 * $i + $j]
            set agt_dst($k)      [new Agent/Null]
            set agt_src($k)      [new Agent/UDP]
            $agt_src($k) set packetSize_ 320
            $agt_src($k) set prio_      0
            $agt_src($k) set fid_       [expr $k + 1]

            set app($k)          [new Application/Traffic/CBR]
            $app($k) set packetSize_ 320
            $app($k) set random_     0
            $app($k) set rate_       128000

            if { $j == 0 } {      ;# enlace descendente
                $ns attach-agent $qap $agt_src($k)
                $ns attach-agent $qsta($i) $agt_dst($k)

                if { $opt(hcca) == 1 } {
                    [$qap getMac 0] tclas [expr $k + 1] 1 [expr $i+1]
                    [$qap getMac 0] tspec 1 [expr $i + 1] downlink
                    [expr 320] [expr 320] 0 .020 .020 0 128000 0
                    .040 1000000 1 0
                }
            } else {              ;# ascendente
                $ns attach-agent $qsta($i) $agt_src($k)
                $ns attach-agent $qap $agt_dst($k)

                if { $opt(hcca) == 1 } {
                    [$qsta($i) getMac 0] tclas [expr $k + 1] 1
                    [$qsta($i) getMac 0] tspec 1 [expr $i + 1]
                    uplink 0 0 0 0 0 0 0 .040 0 1 0
                    [$qap getMac 0] tspec 1 [expr $i + 1] uplink
                    [expr 320] [expr 320] 0 .020 .020 0 128000 0
                    .040 1000000 1 0
                }
            }
        }

        $ns connect $agt_src($k) $agt_dst($k)
        $app($k) attach-agent $agt_src($k)
        $ns at $opt(start) "$app($k) start"
    }
}
```

```
# Fuentes FTP
for { set i 0 } { $i < $opt(m) } { incr i } {
    for { set j 0 } { $j < 2 } { incr j } {
        set k [expr 2 * $i + $j]

        # Agents
        set agt_src_NRT($k) [new Agent/TCP]
        set agt_dst_NRT($k) [new Agent/TCPSink]
        $agt_src_NRT($k) set fid_ [expr 2 * $opt(n) + $k + 1]
        $agt_src_NRT($k) set packetSize_ 1460

        # Applications
        set ftp($k) [new Application/FTP]
        $ftp($k) set packetSize_ 1460
        $ftp($k) set random_ 0

        # Attaching agent to node...
        if { $j == 0 } { ;# downlink
            $ns attach-agent $qap $agt_src_NRT($k)
            $ns attach-agent $qsta([expr $opt(n) + $i])
            $agt_dst_NRT($k)

            if { $opt(hcca) == 1 } {

                # $mac tspec tid qsta direction nomsdu maxsdu
                # maxburst minsi maxsi minrate meanrate maxrate
                # delay minphyrate qack

                [$qap getMac 0] tclas [expr $k + 1] 2 [expr
                    $opt(n) + $i + 1]
                [$qap getMac 0] tspec 2 [expr $opt(n) + $i + 1] \
                    downlink [expr 1500] [expr 1500] 0 .02 .02 0
                    30000 0 .4 1000000 1 0
            }
        } else { ;# enlace ascendente
            $ns attach-agent $qsta([expr $opt(n) + $i])
            $agt_src_NRT($k)
            $ns attach-agent $qap $agt_dst_NRT($k)

            if { $opt(hcca) == 1 } {
                [$qsta([expr $opt(n) + $i]) getMac 0] tclas
                    [expr $k + 1] 2
                #[$qsta([expr $opt(n) + $i]) getMac 0] tspec 2
                    [expr $opt(n) + $i + 1] uplink 1500 1500 0 .01
                    .01 0 30000 0 .04 1000000 1 0
                [$qsta([expr $opt(n) + $i]) getMac 0] tspec 2
                    [expr $opt(n) + $i + 1] uplink 0 0 0 0 0 0 0
                    .40 0 1 0
                [$qap getMac 0] tspec 2 [expr $opt(n) + $i + 1] \
                    uplink [expr 1500] [expr 1500] 0 .02 .02 0 30000
                    0 .4 1000000 1 0
            }
        }
    }
}

# Attaching application to agent...
$ftp($k) attach-agent $agt_src_NRT($k)
```

```
# Conexión de los agentes
$ns connect $agt_src_NRT($k) $agt_dst_NRT($k)
#$ns at $opt(start2) "$ftp($k) start"
#$ns at 80 "$ftp($k) stop"
$ns at $opt(start) "$ftp($k) start"

}

}

#
# Estaciones VOIP
#
for { set i 0 } { $i < $opt(v) } { incr i } {
  for { set j 0 } { $j < 2 } { incr j } {
    set k [expr 2 * $i + $j]

    set source([expr $k]) [new VoipSource]
    $source([expr $k]) model one-to-one

    set encoder([expr $k]) [new Application/VoipEncoder]
    $encoder([expr $k]) codec $opt(codec)

    $source([expr $k]) encoder $encoder([expr $k])

    set decoder([expr $k]) [new
Application/VoipDecoderOptimal]
    $decoder([expr $k]) id [expr $k]
    $decoder([expr $k]) cell-id 0
    $decoder([expr $k]) emodel $opt(codec)

    set agt_src_voip([expr $k]) [new Agent/UDP]
    set agt_dst_voip([expr $k]) [new Agent/UDP]
    $agt_src_voip([expr $k]) set fid_ [expr 2 *
($opt(n)+$opt(m)) + $k + 1]

    if { $j == 0 } { ;# enlace descendente
      $ns attach-agent $qap $agt_src_voip($k)
      $ns attach-agent $qsta([expr $i+$opt(n)+$opt(m)])
        $agt_dst_voip($k)
      if { $opt(hcca) == 1 } {
        [$qap getMac 0] tclas [expr $k + 1] 3
        [expr $opt(n)+$opt(m)+$i + 1]
        [$qap getMac 0] tspec 3
        [expr $i+$opt(n)+$opt(m)+ 1] downlink
        [expr 160] [expr 160] 0 .020 .020 80000 80000 0
        .02 1000000 1 0
      }
    } else { ;# enlace ascendente
      $ns attach-agent $qsta([expr $opt(n)+$opt(m)+$i])
        $agt_src_voip($k)
      $ns attach-agent $qap $agt_dst_voip($k)

      if { $opt(hcca) == 1 } {
        [$qsta([expr $opt(n)+$opt(m)+$i]) getMac 0] tclas
        [expr $k + 1] 1
      }
    }
  }
}
```



```

        [$qsta([expr $opt(n)+$opt(m)+$i]) getMac 0] tspec
        3 [expr $i +$opt(n)+ $opt(m)+1] uplink 0 0 0 0 0
        0 0 0 .060 0 1 0
        [$qap getMac 0] tspec 3
        [expr $i+$opt(n)+$opt(m)+1] uplink [expr 160]
        [expr 160] 0 .010 .010 0 80000 0 .020 1000000
        1 0
    }
}

$ns connect $agt_src_voip([expr $k])
$agt_dst_voip([expr $k])
$encoder([expr $k]) attach-agent
$agt_src_voip([expr $k])$decoder([expr $k])
attach-agent $agt_dst_voip([expr $k])

set aggregate([expr $k]) [new
Application/VoipAggregate]
$aggregate([expr $k]) nframes $opt(aggregate)
$aggregate([expr $k]) attach-agent
$agt_src_voip([expr $k])
$encoder([expr $k]) aggregate $aggregate([expr $k])

# end-to-end modules statistics collection
set tag([expr $k]) [new e2et]
set mon([expr $k]) [new e2em]
if { $opt(tagrand) == "uniform" } {
    set tag_ranvar([expr $k]) [new
RandomVariable/Uniform]
    $tag_ranvar([expr $k]) set min_ 0
    $tag_ranvar([expr $k]) set max_ [expr
$opt(tagmean) / 2]
} elseif { $opt(tagrand) == "exponential" } {
    set tag_ranvar([expr $k]) [new
RandomVariable/Exponential]
    $tag_ranvar([expr $k]) set avg_ $opt(tagmean)
} elseif { $opt(tagrand) == "normal" } {
    set tag_ranvar([expr $k]) [new
RandomVariable/Normal]
    $tag_ranvar([expr $k]) set avg_ $opt(tagmean)
    $tag_ranvar([expr $k]) set std_ [expr $opt(tagvar)
/ 2]
} elseif { $opt(tagrand) == "weibull" } {
    set tag_ranvar([expr $k]) [new
RandomVariable/Weibull]
    $tag_ranvar([expr $k]) set shape_ 2
    $tag_ranvar([expr $k]) set scale_ [expr
$opt(tagmean) / 0.88623]
} elseif { $opt(tagrand) == "constant" } {
    set tag_ranvar([expr $k]) [new
RandomVariable/Constant]
    $tag_ranvar([expr $k]) set val_ $opt(tagmean)
} else {
    puts "Unknown distribution '%s'"
    exit 0
}
}

```

```

        if { $opt(tagrand) != "none" } {
            $tag([expr $k]) ranvar $tag_ranvar([expr $k])
        }
        $tag([expr $k]) per $opt(tagper)
        $agt_src_voip([expr $k]) attach-e2et $tag([expr $k])
        $agt_dst_voip([expr $k]) attach-e2em $mon([expr $k])
        $mon([expr $k]) index [expr $k]
        $mon([expr $k]) start-log
        $ns stat trace voip_frame_delay [expr $k] delay.[expr
        $k]

        $ns at 0.0 "$source([expr $k]) start"
    }
}
#
# Estaciones Video
#
for { set i 0 } { $i < $opt(c) } { incr i } {
#Si j=1 solo funciona el enlace ascendente
    for { set j 0 } { $j < 2 } { incr j } {
        set k [expr 2 * $i + $j]
        # define the trace format:
        Agent/UDP set nam_tracevar_ true
        Agent/UDP set tracevar_ true
        # AGENTES VC
        set agt_src_VC([expr $k]) [new Agent/UDP]
        set agt_dst_VC([expr $k]) [new Agent/Null]
        $agt_src_VC([expr $k]) set fid_ [expr 2 *
        ($opt(n)+$opt(m)+$opt(v))+ $k + 1]
        $agt_dst_VC([expr $k]) set fid_ [expr 2 *
        ($opt(n)+$opt(m)+$opt(v))+ $k + 1]
        set last_time 1799967
    }
# set the simulation end time:
    set end_sim_time [expr 1.0*$last_time/1000+0.001]

# read the video trace file:
    set trace_file [new Tracefile]
    $trace_file filename example-trace-random
    set video([expr $k]) [new Application/Traffic/Trace]
    $video([expr $k]) attach-tracefile $trace_file

    if { $j == 0 } { ;# enlace descendente
        $ns attach-agent $qap $agt_src_VC($k)
        $ns attach-agent $qsta([expr
        $i+$opt(n)+$opt(m)+$opt(v)]) $agt_dst_VC($k)

        if { $opt(hcca) == 1 } {
            [$qap getMac 0] tclas [expr 2 *
            ($opt(n)+$opt(m)+$opt(v))+ $k + 1] 3 [expr $i
            +$opt(n)+ $opt(m)+$opt(v) + 1]
            [$qap getMac 0] tspec 3 [expr $i
            +$opt(n)+$opt(m)+$opt(v)+1] downlink [expr 1540]
            [expr 1540] 0 .016 .016 0 158000 0 .033 1000000 0 1
        }
    } else { ;# enlace ascendente

```

```

        $ns attach-agent $qsta([expr
        $opt(n)+$opt(m)+$opt(v)+$i]) $agt_src_VC($k)
        $ns attach-agent $qap $agt_dst_VC($k)

    if { $opt(hcca) == 1 } {
        [$qsta([expr $opt(n)+$opt(m)+$opt(v)+$i]) getMac
        0] tclas [expr 2 * ($opt(n)+$opt(m)+$opt(v))+
        $k + 1] 3
        [$qap getMac 0] tspec 3 [expr
        $i+$opt(n)+$opt(m)+$opt(v)+1] uplink [expr 1540]
        [expr 1540] 0 .016 .016 0 158000 0 .033 1000000
        0 1
    }
}
$ns connect $agt_src_VC([expr $k]) $agt_dst_VC([expr $k])
$video([expr $k]) attach-agent $agt_src_VC([expr $k])
# Para que cada fuente comience en tiempos distintos
$ns at [expr 10*rand()] "$video([expr $k]) start"
#$video([expr $k]) start
}

}
}
proc finish {} {
    exit 0
}
init
if { $opt(hcca) } {
    $ns at 0.0 status
    $ns at [expr $opt(start) - 0.0001] "hcca-start"
}
$ns run
```

Anexo II .- archivo awk

```
BEGIN {
    recvdSize = 0
    startTime = 0
    stopTime = 400
}
{
    # Trace line format: new
    event = $1
    time = $3
    pkt_id = $41
    pkt_size = 1000
    level = $19

    # Almacena los tiempos de llegada de los paquetes
    if (sendTime[pkt_id] == 0 && (event == "+" || event == "s")
    && pkt_size >= pkt) {
        startTime = startTime+0
        if (time < startTime) {
            startTime = time
        }
        sendTime[pkt_id] = time
    }
    # Actualización del número total de paquetes recibidos
    if (event == "r" && pkt_size >= pkt) {

        if (time > stopTime) {
            stopTime = time
        }
        # Rip off the header
        hdr_size = pkt_size % pkt
        pkt_size -= hdr_size
        # Store received packet's size
        recvdSize += pkt_size
        # Store packet's reception time
        recvTime[pkt_id] = time
    }
}
END {
    # Cálculo del retardo medio
    delay = avg_delay = recvdNum = sendNum=0
    for (i in recvTime) {
        if (sendTime[i] == 0) {
            printf("\nError in delay.awk: receiving a packet
            that wasn't sent %g\n",i)
        }

        delay2[i]= recvTime[i] - sendTime[i]
        delay += recvTime[i] - sendTime[i]
        recvdNum ++
    }
}
```

```
#Cálculo del percentil

numeroElementos_delay=asort(delay2,delay2ordenado)
#Da el número de elementos de delay2 y no ordena
sort(delay2ordenado,numeroElementos_delay)
#Ordena delay2ordenado

#for (i=1;i<numeroElementos_delay;i++) {
    #printf(" %15s:  %g\n", "avgDelay[ms]",
        delay2ordenado[i]*1000)
#}
percentil=int(99*numeroElementos_delay/100)
percentil_final=delay2ordenado[percentil];
#####
#####
for (i in sendTime) {
    sendNum ++
}
#####
#####
if (recvdNum != 0) {
    avg_delay = delay / recvdNum
} else {
    avg_delay = 0
}
# Output
if (recvdNum == 0) {

printf("#####\n" \
      "# Warning: no packets were received,
simulation may be too short #\n" \

"#####\n\n")
}
printf(" %15s:  %g\n", "receivedPkts", recvdNum)
printf(" %15s:  %g\n", "sendPkts",sendNum)
printf(" %15s:  %g\n", "avgTput[kbps]",
(recvdSize/(stopTime-startTime))*(8/1000))
printf(" %15s:  %g\n", "avgDelay[ms]", avg_delay*1000)
printf(" %15s:  %g\n", "Percentil", percentil_final*1000)

}
function abs(value) {
    if (value < 0) value = 0-value
    return value
}
function sort (A,n){
    for (i=1;i<=n;i++){
        A[i]=A[i]+0
    }
    for (i=2;i<=n;++i){
        for (j=i; (j>1) in A && A[j-1]>A[j]; --j) {
            tmp=A[j]
            A[j]=A[j-1]

```

```
        A[j-1]=tmp
    }
}
return
}
```

Anexo III.- archivo Matlab: ASCII → binario

```
A=fopen('verbose_33.txt','r');

% Vuelca los datos del fichero en un vector (e.g. en caso de
% números enteros =>'%d' / '%f'=> para flotantes)

datos=fscanf(A,'%d',[2 inf]);

datos=int32(datos);

% Esta función permite que los números se puedan convertir a
% big endian, condición necesaria para que NS2 pueda usar esta
% traza

datos=swapbytes(datos);

fclose(A);

% Abre un nuevo fichero con permisos de escritura binaria

V=fopen('example-trace','w');

% Escribe los datos en el fichero en formato binario tipo
% 'int32'(aunque podría ser 'int16',...o cualquier otro)

fwrite(V,datos,'int32');

fclose(V);
```