

Trabajo Fin de Grado  
Grado Ingeniería de las Tecnologías de  
Telecomunicación

## EVALUACIÓN Y OPTIMIZACIÓN DEL CONSUMO DE RECURSOS EN VoWIFI

Autor: Miguel Fontanilla Pérez de Tudela

Tutor: Rafael Estepa Alonso

Dep. de Ingeniería Telemática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2016





Trabajo Fin de Grado  
Grado Ingeniería de las Tecnologías de Telecomunicación

# **EVALUACIÓN Y OPTIMIZACIÓN DEL CONSUMO DE RECURSOS EN VoWIFI**

Autor:

Miguel Fontanilla Pérez de Tudela

Tutor:

Rafael Estepa Alonso

Profesor titular

Dep. de Telemática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016



Autor: Miguel Fontanilla Pérez de Tudela

Tutor: Rafael Estepa Alonso

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal



*A mi familia*

*A mis maestros*

*Y con un especial significado, a  
mi Padre*





# Agradecimientos

---

Los dispositivos empleados para el desarrollo de este proyecto han sido cedidos por el departamento de Ingeniería Telemática de la Escuela Técnica Superior de Ingeniería de Sevilla. Así mismo se ha realizado la compra de varios adaptadores inalámbricos para poder continuar con el desarrollo del proyecto que han sido financiados por el departamento, que también me cedió un laboratorio para poder llevar a cabo las pruebas de los dispositivos y desarrollar escenarios para la experimentación. Por ello les estoy muy agradecido.

Mi tutor, Rafael Estepa Alonso, se ha implicado mucho en el proyecto y, gracias a una constante comunicación y a sus consejos, el proyecto ha podido avanzar, por lo que también quisiera mostrarle mi agradecimiento.

*Miguel Fontanilla Pérez de Tudela*

*Sevilla, 2016*



# Resumen

---

La tecnología de voz sobre protocolo de internet o VoIP ha sufrido un gran desarrollo durante los últimos años y por ello se ha convertido en un objeto de estudio; especialmente en las cuestiones relacionadas con la calidad de servicio ofrecida.

Originalmente los teléfonos empleados eran dispositivos cableados, pero debido al desarrollo de las tecnologías inalámbricas, la voz sobre IP en redes inalámbricas ha tomado un peso importante.

La utilización de un medio inalámbrico implica una serie de restricciones derivadas del empleo de un medio físico común y saturado, y de terminales con baterías.

El objetivo de este documento es el desarrollo de un estudio de los mecanismos de calidad de servicio y ahorro de energía empleados en redes inalámbricas y la realización de pruebas que permitan valorar el funcionamiento de los mismos.



# Abstract

---

VoIP technology has experienced a great development during the last years and as a result it has become an object of study; especially on issues related to quality of service.

Originally, telephones were wired devices, but due to the development of wireless technologies, voice over IP over wireless networks has taken an important weight.

The use of a wireless medium involves a number of restrictions arising from the use of a common and saturated physical medium, and non-wired terminals.

The purpose of this paper is the development of a study of the mechanisms of quality of service and power saving used in wireless networks, and to test these mechanisms.

# Índice

---

|  |             |
|--|-------------|
| <b>Agradecimientos</b>                                   | <b>ix</b>   |
| <b>Resumen</b>   | <b>xi</b>   |
| <b>Abstract</b>  | <b>xiii</b> |
| <b>Índice</b>  | <b>xiv</b>  |
| <b>Índice de Tablas</b>                                  | <b>xvi</b>  |
| <b>Índice de Figuras</b>                                 | <b>xvii</b> |
| <b>1 Introducción y objetivos</b>                        | <b>1</b>    |
| <b>2 VoIP,Wi-Fi y PSM</b>                                | <b>5</b>    |
| 2.1 <i>Funcionamiento básico de VoIP</i>                 | 5           |
| 2.1.1 Calidad de Servicio en VoIP                        | 6           |
| 2.1.2 Señalización en VoIP                               | 7           |
| 2.1.3 RTP/RTCP   | 8           |
| 2.2 <i>Funcionamiento de Wi-Fi</i>                       | 9           |
| 2.2.1 Estándares 802.11                                  | 9           |
| <b>3 Pruebas e Investigación</b>                         | <b>23</b>   |
| 3.1 <i>Dispositivos Wi-Fi</i>                            | 23          |
| 3.1.1 Edimax EW7811                                      | 23          |
| 3.1.2 TP-Link TL-WN725N                                  | 23          |
| 3.1.3 D-Link DWA-160 rev.A2                              | 23          |
| 3.1.4 D-Link DWA-160 rev.C1                              | 24          |
| 3.1.5 D-Link DAP 2590                                    | 25          |
| 3.1 <i>Pruebas en sistemas operativos (Primera fase)</i> | 26          |
| 3.1.1 Linux  | 26          |
| 3.1.2 Windows  | 29          |
| 3.2 <i>Pruebas en sistemas operativos (Segunda Fase)</i> | 34          |
| 3.2.1 Windows 7  | 36          |
| <b>4 Pruebas Sobre Escenarios Reales</b>                 | <b>45</b>   |
| 4.1 <i>El escenario</i>                                  | 45          |
| 4.1.1 Red de sincronismo                                 | 46          |
| 4.1.2 Red de datos                                       | 49          |
| 4.2 <i>PJSIP</i>   | 49          |
| 4.3 <i>Pruebas</i>                                       | 51          |

|          |  |           |
|----------|--|-----------|
| 4.3.1    | Terminales VoIP                              | 52        |
| 4.3.2    | Sniffer                                      | 53        |
| <b>5</b> | <b>Análisis de los resultados</b>            | <b>54</b> |
| 5.1      | <i>Resultados de las pruebas</i>             | 54        |
| 5.1.1    | Software para el análisis: UAPSD             | 54        |
| 5.1.2    | Medidas de QoS                               | 64        |
| 5.2      | <i>Estadísticas obtenidas</i>                | 68        |
| 5.2.1    | Modelo-E                                     | 68        |
| 5.2.2    | P.563  | 72        |
| 5.2.3    | P.862  | 72        |
| 5.2.4    | Energía y periodos de sueño                  | 73        |
| 5.2.5    | Ahorro, calidad y códecs                     | 78        |
| 5.2.6    | Conclusiones                                 | 79        |
| 5.2.7    | Líneas de futuras de trabajo                 | 81        |
|          | <b>Referencias</b>                           | <b>83</b> |
|          | <b>Anexo A: Configuración de los equipos</b> | <b>86</b> |
|          | <b>Anexo B: Scripts y código fuente</b>      | <b>92</b> |
|          | <b>Glosario</b>                              | <b>96</b> |

# ÍNDICE DE TABLAS

---

|  |    |
|--|----|
| Tabla 3-1. Comparativa de los adaptadores de red                 | 24 |
| Tabla 3-2. Comparativa de los adaptadores de red                 | 34 |
| Tabla 4-1. Comparativa de los códecs empleados                   | 51 |
| Tabla 5-1. Valores de $I_e$ y $B_{pl}$ para los distintos códecs | 67 |
| Tabla 5-2. Porcentaje de ahorro de energía para G.711 con UAPSD  | 75 |
| Tabla 5-3. Porcentaje de ahorro de energía para iLBC con UAPSD   | 76 |
| Tabla 5-4. Porcentaje de ahorro de energía para Speex con UAPSD  | 77 |



# ÍNDICE DE FIGURAS

---

|   |    |
|---|----|
| Figura 1-1.Red corporativa con interconexión a PSTN                             | 1  |
| Figura 1-2. Red VoWLAN con interconexión a PSTN [1]                             | 2  |
| Figura 2-1. Torre de protocolos VoIP [3]  | 6  |
| Figura 2-2. Intercambio de mensajes con un SIP Proxy [4]                        | 8  |
| Figura 2-3. Arquitecturas de las capas OSI 1 y 2 del estándar 802.11 [8]        | 10 |
| Figura 2-4. Arquitectura de una red 802.11 [9]                                  | 11 |
| Figura 2-5. Formato de trama física en 802.11b [5]                              | 12 |
| Figura 2-6. Formatos de trama MAC en 802.11 [5]                                 | 14 |
| Figura 2-7. Esquema simplificado de CSMA/CA [5]                                 | 15 |
| Figura 2-8. Acceso básico en DCF empleando CSMA/CA [6]                          | 15 |
| Figura 2-9. Envío con reserva previa [6]  | 16 |
| Figura 2-10. Mecanismo de Backoff con varias estaciones [5]                     | 16 |
| Figura 2-11.Formato genérico de una trama Beacon [7]                            | 17 |
| Figura 3-1. De izquierda a derecha EW7811Un, TL-WN725N, DWA-160 A2 y DWA-160 C1 | 24 |
| Figura 3-2. D-Link DAP-2590 [25]  | 26 |
| Figura 3-3. Ejemplos de comandos con iwconfig.                                  | 27 |
| Figura 3-4. Algunos Ioctls privados del adaptador EW7811Un.                     | 28 |
| Figura 3-5. Configuración avanzada del adaptador inalámbrico                    | 29 |
| Figura 3-6. Consola de gestión de Microsoft                                     | 30 |
| Figura 3-7. Mapeo de DSCP/COS/WMM AC [17]                                       | 30 |
| Figura 3-8. Editando la configuración de un tipo de servicio                    | 31 |
| Figura 3-9. Topología adoptada en la prueba                                     | 32 |
| Figura 3-10. Detalle de un Beacon emitido por el AP                             | 32 |
| Figura 3-11. Mapeo correcto DSCP/AC   | 33 |
| Figura 3-12. Detalle de un Association Request                                  | 33 |
| Figura 3-13. Detalle de un Association Request                                  | 34 |
| Figura 3-14. Flags de una cabecera 802.11                                       | 34 |

|   |    |
|---|----|
| Figura 3-15. Detalle del Campo QoS Control de una cabecera 802.11               | 34 |
| Figura 3-16. Asus PCE-AC56 con soporte de antenas                               | 36 |
| Figura 3-17. Detalle del asistente de configuración de Asus                     | 37 |
| Figura 3-18. Detalle del interfaz de configuración de Windows                   | 37 |
| Figura 3-19. Detalle de un Association Request                                  | 38 |
| Figura 3-20. Detalle los registros del adaptador PC-AC56                        | 38 |
| Figura 3-21. Asus PCE-AC68 con soporte de antenas.                              | 38 |
| Figura 3-22. Intel AC-7260 (Housing de fenvi)                                   | 39 |
| Figura 3-23. Propiedades del AC-7260 en el interfaz de configuración de Windows | 40 |
| Figura 3-24. Detalle de un Association Request                                  | 40 |
| Figura 3-25. Secuencia de mensajes analizados en la captura                     | 41 |
| Figura 3-26. Detalle de los flags de una cabecera 802.11                        | 41 |
| Figura 3-27. Detalle de los flags de una cabecera 802.11                        | 41 |
| Figura 3-28. Detalle del campo QoS Control de una cabecera 802.11               | 41 |
| Figura 3-29. Detalle de los flags de una cabecera 802.11                        | 42 |
| Figura 3-30. Detalle del campo QoS Control de una cabecera 802.11               | 42 |
| Figura 3-31. Esquema del funcionamiento de WMM-PS observado                     | 42 |
| Figura 4-1. Esquema del escenario   | 45 |
| Figura 4-2. Escenario planteado orginamente                                     | 46 |
| Figura 4-3. Esquema simplificado del intercambio de mensajes PTP [18]           | 47 |
| Figura 4-4. Esquema de la red utilizada para las pruebas                        | 49 |
| Figura 4-5. Consola de comandos de PJSUA  | 50 |
| Figura 4-6. Ejemplo de fichero de configuración                                 | 52 |
| Figura 5-1. Primer estado   | 56 |
| Figura 5-2. Detalle de un beacon del AP: valores de los parámetros de EDCA      | 56 |
| Figura 5-3. Intercambio de mensajes con varios paquetes ascendentes en TXOP     | 57 |
| Figura 5-4. Segundo estado  | 57 |
| Figura 5-5. Tercer estado   | 58 |
| Figura 5-6. Cuarto estado   | 59 |
| Figura 5-7. Quinto estado   | 60 |
| Figura 5-8. Patrón del quinto estado observado en una captura                   | 60 |
| Figura 5-9. Relaciones entre los estados  | 61 |
| Figura 5-10. Relaciones entre los estados (sin UAPSD)                           | 62 |
| Figura 5-11. Escala MOS [22]  | 65 |
| Figura 5-12. MOS obtenida frente a Nfpp del canal descendente.                  | 69 |
| Figura 5-13. MOS obtenida frente a Nfpp del canal acendente.                    | 69 |
| Figura 5-13. MOS obtenida frente a Nfpp del canal descendente.                  | 70 |
| Figura 5-14. MOS obtenida frente a Nfpp del canal ascendente.                   | 70 |
| Figura 5-15. MOS obtenida frente a Nfpp del canal descendente.                  | 71 |
| Figura 5-16. MOS obtenida frente a Nfpp del canal ascendente.                   | 72 |

|  |    |
|--|----|
| Figura 5-17. Periodos totales de sueño y duración media del periodo frente a Nfpp. | 74 |
| Figura 5-18. Tiempo total de sueño frente a Nfpp.                                  | 74 |
| Figura 5-19. Periodos totales de sueño y duración media del periodo frente a Nfpp  | 75 |
| Figura 5-20. Tiempo total de sueño frente a Nfpp                                   | 76 |
| Figura 5-21. Periodos totales de sueño y duración media del periodo frente a Nfpp  | 77 |
| Figura 5-22. Tiempo total de sueño frente a Nfpp                                   | 77 |
| Figura 5-23. Mejor Códec,Nfpp en función del ahorro para una MOS mínima            | 79 |
| Figura A-1. Detalle del Domain Time Applet   | 86 |
| Figura A-2. Intercambio de mensajes PTP de sincronización                          | 87 |
| Figura A-3. Ejemplo de arranque del servidor ptpd2                                 | 88 |
| Figura A-4. Esquema de la red utilizada para las pruebas de sincronismo            | 88 |
| Figura A-5. Latencia del PC1 medida por Domain Time II                             | 89 |
| Figura A-6. Latencia del PC2 medida por Domain Time II                             | 89 |
| Figura A-7. Interfaz de configuración web del DAP-2590                             | 90 |



# 1 INTRODUCCIÓN Y OBJETIVOS

*Projects we have completed demonstrate what we know, future projects decide what we will learn.*

Dr. Mohsin Tiwana

## 1.1 Introducción

El intercambio de voz en tiempo real a través de una red de ordenadores y, en especial a través de Internet es uno de los servicios más demandados y estudiados en los últimos años. Este servicio está estrechamente asociado al término VoIP (Voice over Internet Protocol). Cuando hablamos de VoIP nos referimos de manera general al sistema de transporte de voz digitalizada a través de redes de datos basadas en el protocolo IP (RFC971).

La telefonía sobre IP ha sido ampliamente aceptada y desplegada en empresas de todo el mundo, pues permite reutilizar las conexiones de datos de los puestos de trabajo de los empleados para realizar llamadas telefónicas. Esta arquitectura permite cursar llamadas internas entre los terminales conectados a la red de una empresa, desbancando así a los anteriores teléfonos analógicos o con interfaz RDSI.

Además del tráfico de voz a nivel interno, VoIP permite realizar llamadas a líneas externas; bien a través de internet o mediante gateways o pasarelas que permiten transportar la voz digitalizada sobre circuitos de voz tradicionales y traducen la señalización empleada por VoIP al sistema de señalización de la red telefónica conmutada (PSTN).

En la siguiente figura mostramos un ejemplo de un posible escenario genérico de VoIP para empresa, con salida a internet y un Gateway para la interconexión con la red telefónica conmutada:

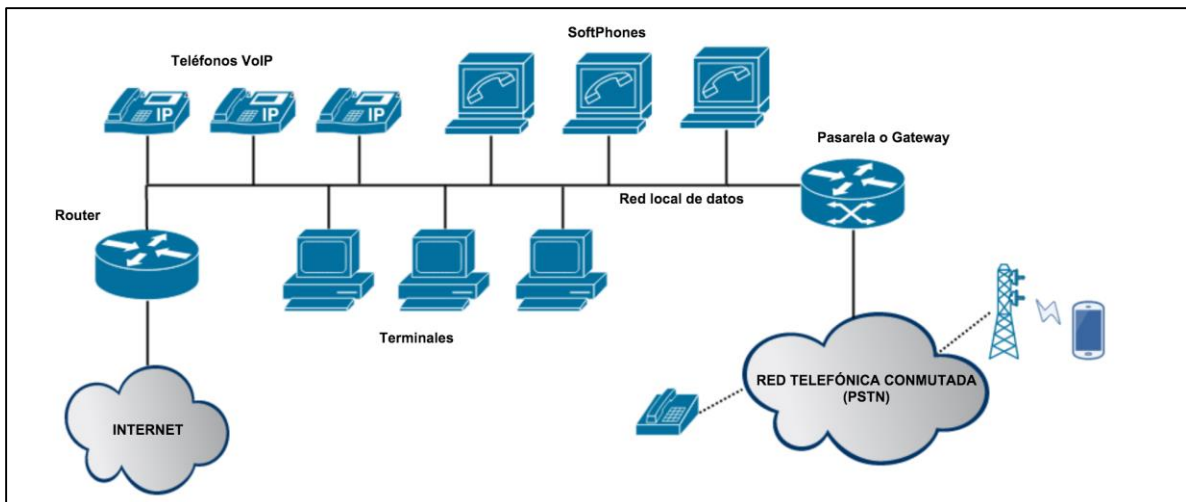


Figura 1-1. Red corporativa con interconexión a PSTN

La ventaja principal que ofrece la implantación de soluciones de VoIP como la mostrada es evidente: la red de datos previamente implantada se emplea para soportar tanto tráfico de datos como tráfico de voz, unificando así en una única red todos los servicios. No obstante, es necesario señalar, que el dimensionamiento de la red debe ser adecuado, pues la telefonía sobre IP presenta unos requisitos de retardo y pérdidas que deben ser respetados para su correcto funcionamiento.

Esta primera aproximación a la infraestructura de VoIP se ha visto modificada en los últimos años con la proliferación y el desarrollo de las tecnologías de redes de área local inalámbricas o Wireless LAN (WLAN). La modificación consiste básicamente en la sustitución de los terminales VoIP cableados por terminales inalámbricos, lo cual permite la movilidad de los mismos. Estos terminales se conectan mediante protocolos de red inalámbrica definidos en los estándares IEEE 802.11 y posteriores revisiones, bien directamente al encaminador, en este caso equipado con una interfaz inalámbrica, o a un punto de acceso que finalmente entregaría los paquetes al encaminador.

Para designar a este tipo de arquitectura, se suele emplear el término VoWLAN (Voice over Wireless LAN), y los dispositivos que pueden emplearse como terminales de usuario son todos aquellos con conexión inalámbrica a internet, como ordenadores, teléfonos IP-DECT o más recientemente smartphones.

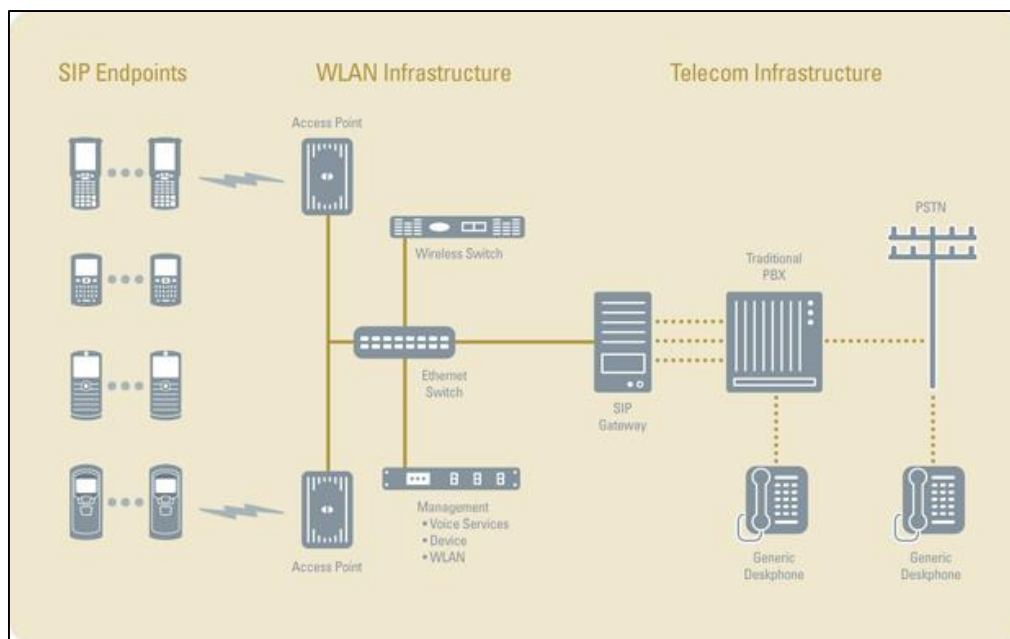


Figura 1-2. Red VoWLAN con interconexión a PSTN [1]

Esta modificación implica una serie de aspectos que se deben tener en cuenta:

- El espectro radioeléctrico es un recurso muy limitado por lo que es necesario gestionar el ancho de banda, y por ende el número máximo de dispositivos que pueden conectarse, y las bandas ISM (Industrial, Scientific and Medic) empleadas por los protocolos del estándar IEEE 802.11 (entre otros) suelen estar muy contaminadas por otras señales.
- En el caso de redes empresariales de una dimensión considerable, en las que se empleen varios puntos de acceso, es necesario gestionar el acceso, la movilidad de los usuarios y el traspaso entre los diferentes puntos de acceso, lo cual puede tener una repercusión sobre la calidad y continuidad de las llamadas.
- Los terminales ya no se encontrarán alimentados de forma continua mediante la red de comunicaciones (PoE) o mediante una conexión a la red eléctrica, sino que dependerán de baterías recargables, por lo que el consumo energético se convierte en un factor determinante.

Este trabajo se centrará, por tanto, en el estudio del consumo energético que realizan los terminales inalámbricos, intentando minimizarlo, para aumentar así la autonomía de los terminales. Se investigarán los distintos modos de ahorro de energía que se definen en el estándar IEEE 802.11 y el posterior IEEE 802.11e, y los parámetros más relevantes de cada uno de ellos.

El estudio que nos ocupará parte de un Trabajo de fin de Carrera anterior, realizado en 2013 por Daniel Jiménez Lara, alumno de la Escuela Técnica Superior de Ingeniería de Sevilla, y centrado en el estudio experimental de calidad y consumo energético de VoIP sobre redes 802.11 [2].

Es necesario señalar que el cómputo del consumo energético en este trabajo se realizó de forma teórica y basando los cálculos en el modo de ahorro de energía definido en el estándar 802.11 original, mientras que el

objeto de este trabajo será obtener estadísticas experimentales del consumo energético y su relación con los distintos códecs de VoIP empleados, utilizando diferentes métodos de ahorro de energía.

## 1.2 Objetivos

El correcto funcionamiento de VoIP exige, como se ha comentado, el cumplimiento de unos determinados niveles mínimos de calidad. La calidad de las llamadas en VoIP está estrechamente relacionada con las pérdidas y los diferentes retardos que experimentan los paquetes durante la comunicación, ya sea por retardos que introduce la red o por retardos debidos al procesamiento de los paquetes en los extremos. Por tanto, es crucial conocer el retardo y el porcentaje de pérdidas sufridos por los paquetes para poder actuar en consecuencia. El proyecto que se adopta como punto de partida para este proporciona un algoritmo que permite modificar el número de tramas de voz por paquete de forma adaptativa con el objetivo de mantener un determinado nivel de calidad de servicio, así como de optimizar el consumo energético.

No puede olvidarse tampoco el problema adicional que supone el empleo de terminales móviles como equipamiento de usuario, pues no disponen de conexión a la red eléctrica de forma continua, sino que se alimentan a través de baterías, lo que convierte al consumo energético en una cuestión fundamental. Las antenas de los dispositivos móviles, más concretamente, los transceptores de las mismas, tienen un alto impacto en el consumo energético, tanto en transmisión como en recepción, por lo que la duración de los intervalos en los que las antenas están transmitiendo, recibiendo información o en un estado ocioso (incluso de sueño), y por tanto consumiendo energía, será crítica en el desarrollo de este proyecto.

De este modo, el objetivo del presente proyecto es evaluar la eficiencia de los mecanismos de ahorro de energía existentes para las redes 802.11, especialmente aquellos que han sido concebidos para aplicaciones sensibles al retardo, con el objetivo de garantizar una calidad determinada para el servicio de telefonía sobre IP, con diferentes códecs y diferentes valores para el número de tramas por paquete empleados por dichos códecs.

La introducción de periodos de inactividad durante la comunicación puede penalizar la calidad de las llamadas, por lo que será necesario buscar un punto de equilibrio entre el ahorro energético y la calidad de servicio ofrecida.

Para la realización del proyecto plantearemos una serie de puntos a seguir:

- 1) Análisis de los distintos modos de ahorro de energía que se definen en el estándar 802.11 y revisiones de la misma serie.
- 2) Análisis de los drivers de varios adaptadores de redes de área local inalámbrica en Linux y Windows.
- 3) Pruebas de funcionamiento de los drivers y adaptadores de red inalámbricos para la verificación de los diferentes modos de ahorro de energía previamente planteados.
- 4) Instalación del software pjsip con las modificaciones que se le incluyeron en el trabajo que se toma como punto de partida
- 5) Planteamiento de los escenarios de prueba a utilizar.
- 6) Configuración de los escenarios y medidas del retardo.
- 7) Realización de las pruebas experimentales.
- 8) Análisis de los resultados obtenidos.
- 9) Desarrollo de un software adecuado para el cálculo de la energía consumida y la obtención de estadísticas a partir de los resultados.
- 10) Búsqueda de un punto de equilibrio entre el ahorro energético y la calidad de servicio obtenida.
- 11) Conclusiones y posibles líneas de mejora





# 2 VoIP, Wi-Fi y PSM

---

*An expert is a man who has made all the mistakes, which can be made, in a very narrow field.*

Niels Bohr

**E**n la investigación que nos ocupa intervienen diferentes tecnologías y estándares de telecomunicación:

VoIP, Wi-Fi y los diferentes mecanismos de ahorro de energía para redes de área local inalámbricas, por lo que se dedicará este capítulo al análisis de los mismos y sus características.

Se analizará cuáles son las características apropiadas que debe tener una red de conmutación de paquetes para que la voz se pueda transmitir de forma correcta. No se entrará en detalle respecto a los formatos de paquete en VoIP, tanto de datos como de señalización.

De la misma forma se profundizará en los diferentes estándares de la serie 802.11 en los que se define el uso de los niveles 1 y 2 del modelo OSI para las redes de área local inalámbricas (WLAN), y empleo de los mecanismos de ahorro de energía en cada uno de ellos.

## 2.1 Funcionamiento básico de VoIP

Voz sobre IP es una tecnología que permite realizar llamadas telefónicas sobre redes de datos, tanto internet como redes de área local. La telefonía tradicional se basa en el transporte de la señal analógica mediante conmutación de circuitos, pero en VoIP se convierte la voz analógica en paquetes de datos digitales que soportan la comunicación sobre el protocolo IP y UDP (User Datagram Protocol/RFC768). [3] Además, pueden emplear protocolos para aplicaciones en tiempo real como RTP (Real Time Protocol/RFC3550). [4]

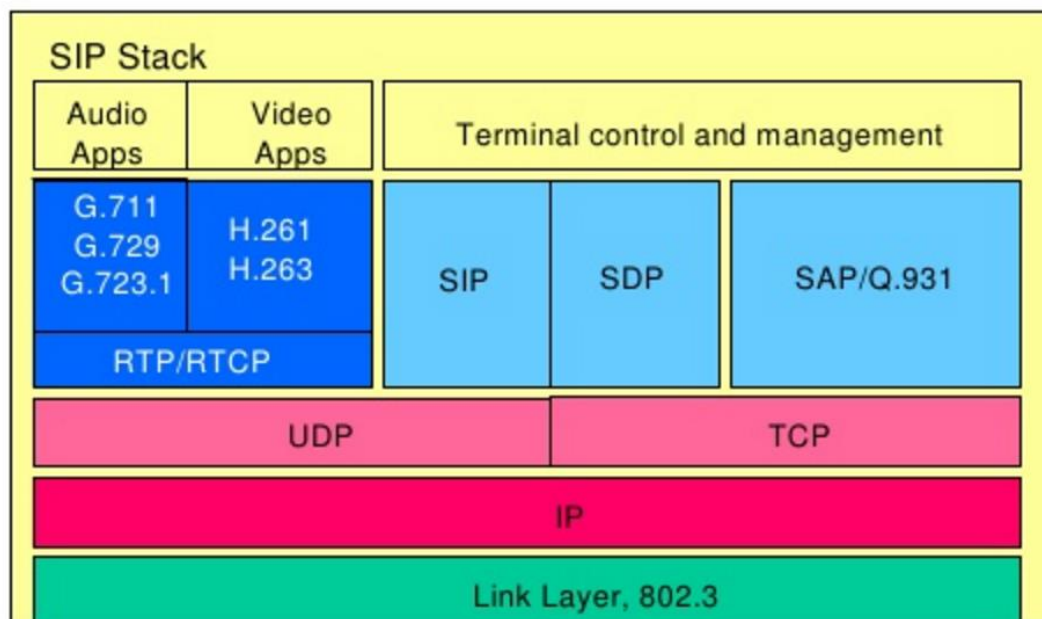


Figura 2-1. Torre de protocolos VoIP [3]

Como cualquier otra tecnología, VoIP también posee ciertos defectos:

- Durante una comunicación con VoIP puede llegar a producirse retraso en la llegada de los paquetes o pérdidas que produzcan un deterioro de la calidad percibida por el usuario.
- Uno de los requisitos más importantes para aportar calidad de servicio a una aplicación de VoIP es proporcionar suficiente ancho de banda a la red para que pueda soportar el tráfico de voz en tiempo real.

### 2.1.1 Calidad de Servicio en VoIP

Para la medición de la calidad de servicio ofrecida por VoIP existen cuatro parámetros fundamentales: el ancho de banda, retardo temporal o latencia (delay), variación de retardo (jitter) y pérdida de paquetes.

Los factores principales de los que depende el ancho de banda son el codificador o códec empleado y el número de muestras o tramas de voz que se deseen introducir por paquete. Esto es así ya que ambos factores están directamente relacionados con el retardo. Los codificadores en los terminales de VoIP comprimen la señal de voz e introducen dos tipos de retraso:

- procesamiento: tiempo necesario para codificar el código en una sola trama de voz.
- retraso de la trama: tiempo necesario para que el sistema transmita la trama.

Existen tres tipos de codificadores o códecs: Los de forma de onda, los de fuente y los híbridos. Los de forma de onda consiguen una alta calidad de sonido pero consumen mucho ancho de banda, mientras que los de fuente consumen un ancho de banda menor, pero ofrecen una calidad de sonido menor. Los híbridos intentan proporcionar un equilibrio entre las características de los codificadores de forma de onda y los de fuente.

Entre los codificadores utilizados en VoIP encontramos los G.711 G.723.1 y el G.729 (especificados por la ITU-T), que pueden observarse en la torre de protocolos de la figura 2-1.

El jitter se define como la variación en el tiempo de llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino.

El jitter entre el punto inicial y final de la comunicación debe ser inferior a 100 ms. Si el valor es menor a 100 ms el jitter puede ser compensado de manera apropiada. En caso contrario debe ser minimizado.

La solución más ampliamente adoptada es la utilización del buffer de jitter, que consiste básicamente en asignar una pequeña cola para ir recibiendo los paquetes y sirviéndolos con un pequeño retraso. Si algún paquete no está en el buffer (se perdió o no ha llegado todavía), se descarta cuando sea necesario.

Normalmente en los teléfonos IP se puede modificar el tamaño del buffer. Un aumento del buffer implica una menor pérdida de paquetes, pero genera un retardo mayor. Una disminución del tamaño del buffer implica menos retardo, pero más una mayor pérdida de paquetes.

La latencia o delay se define como el tiempo que tarda un paquete en llegar desde la fuente al destino. La latencia o retardo entre el punto inicial y final de la comunicación debe ser inferior a 250 ms. El oído humano es capaz de detectar latencias de unos 250 ms, 200 ms en el caso de personas bastante sensibles. Si se supera ese umbral la comunicación se vuelve molesta.

No hay una solución que se pueda implementar de manera sencilla. Muchas veces depende de los equipos por los que pasan los paquetes. Se puede intentar reservar un ancho de banda de origen a destino o señalar los paquetes con valores de TOS (campo de la cabecera IPv4 para priorización) para que los equipos los identifiquen como tráfico en tiempo real y lo traten con mayor prioridad, pero no suelen ser medidas muy eficaces si los paquetes tienen que atravesar redes sobre las que no se tiene control.

La pérdida de paquetes es otro de los factores determinantes, ya que puede ocurrir por muchas razones, y en algunos casos es inevitable, aunque la principal causa es la congestión de la red. Durante los procesos de congestión los routers y switches pueden sobrepasar sus colas y descartar paquetes. Otra causa que provoca la pérdida de paquetes es un exceso de latencia, esto es: un grupo de paquetes llega tarde y deben ser descartados en favor de los nuevos.

Los paquetes perdidos de las aplicaciones que no son de tiempo real no son críticos. Estas aplicaciones suelen usar el protocolo TCP (Transmisión Control Protocol)[6] que permite retransmisión, no supondría un gran problema. Sin embargo, las aplicaciones basadas en tiempo real como UDP son menos tolerantes a la pérdida de paquetes porque no tienen posibilidad de retransmisión.

La consecuencia que provoca la pérdida de paquetes es que se producen vacíos en la conversación. La ventaja que tienen los paquetes de VoIP es que su longitud es corta, la pérdida de esta pequeña cantidad de diálogo no empeora la conversación de los interlocutores. Pero probabilísticamente la pérdida de un paquete significa la pérdida de varios paquetes, este hecho sí degrada severamente la calidad del servicio de una red VoIP. Este fenómeno es conocido como pérdidas en ráfagas (burst).

Se determina que un porcentaje tolerable de pérdida de paquetes se centra entre 1% y 5% y la calidad empieza a ser intolerable cuando más del 5% de los paquetes de voz se han perdido. [2]

### 2.1.2 Señalización en VoIP

Session Initiation Protocol (SIP/RFC3261) [7] es el estándar del IETF que se utiliza para establecer las conexiones en VoIP. Se trata de un protocolo de control de la capa de aplicación encargado de crear, modificar y terminar sesiones con uno o más participantes. Estas sesiones multimedia incluyen aplicaciones de audio, video que pueden utilizar flujos unicast (punto a punto) o multicast (multipunto).

El protocolo SIP se vale de las funciones aportadas por otros protocolos, que da por implementadas y no vuelve a desarrollar. Debido a este concepto, SIP funciona en colaboración con otros muchos protocolos. El protocolo SIP se concentra en el establecimiento, modificación y terminación de las sesiones, y se complementa entre otros con Session Description Protocol (SDP), que describe el contenido multimedia de la sesión, por ejemplo qué direcciones IP, puertos y códecs se usarán durante la comunicación.

Se complementa también con Real-time Transport Protocol (RTP), que posteriormente se analizará, portador del contenido de voz y vídeo que intercambian los participantes en una sesión establecida por SIP.

Los mensajes que se intercambian en el protocolo SIP pueden ser peticiones o respuestas. Las peticiones tienen una línea de petición, una serie de encabezados y un cuerpo. En la línea de petición se indica el propósito de la petición y el destinatario de la petición.

Las peticiones tienen distintas funciones. El propósito de una petición está determinado por lo que se denomina el Método de dicha petición, que no es más que un identificador del propósito de la petición.

Las respuestas se generan como retorno de una petición devolviendo un código de estado, similares a los empleados por el protocolo HTTP (RFC 2616) [8]. En este caso la línea inicial recibe el nombre de status line, que llevara el mensaje SIP utilizado, código de respuesta y una pequeña descripción de ese código. Podemos recibir estas respuestas según el rango.

El cuerpo de los mensajes es opcional y se utiliza, entre otras cosas, para transportar las descripciones de las

sesiones que se quieren establecer, utilizando la sintaxis del protocolo SDP.

La localización la basa en Sistemas de Nombre de Dominio (DNS/RFC 1035) [9]. En la figura 2-2 se puede ver el intercambio de mensajes de señalización durante una llamada de VoIP.

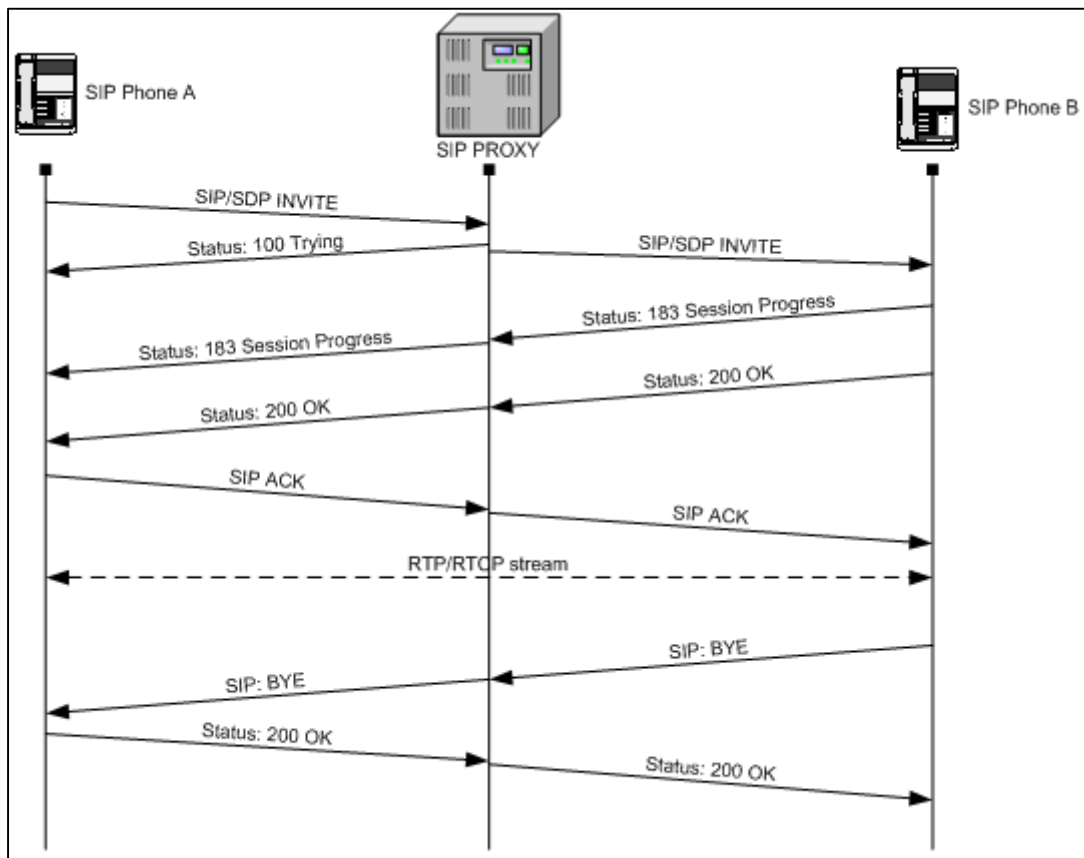


Figura 2-2. Intercambio de mensajes con un SIP Proxy [4]

### 2.1.3 RTP/RTCP

Para transportar la voz sobre IP, se utilizan el protocolo IP a nivel de red y el protocolo UDP a nivel de transporte, como se puede observar en la figura 2-1. Pero estos dos protocolos no son suficientes para asegurar el transporte de la voz. De hecho, UDP es un protocolo sin corrección de errores, y en ningún momento se asegura la llegada de paquetes en su orden de emisión. Para el transporte de datos en tiempo real, como la voz o el vídeo, es necesario utilizar dos protocolos suplementarios: RTP (Real-Time Transport Protocol) y RTCP (RTP Control Protocol) [4].

RTP y RTCP son dos protocolos que se sitúan en el nivel de aplicación y se utilizan junto con el protocolo de transporte UDP. RTP y RTCP pueden utilizar el modo unicast (punto a punto) y el modo multicast (multipunto). RTP y RTCP utilizan puertos diferentes. RTP utiliza un número de puerto par, y RTCP el número de puerto impar que sigue a continuación. Cuando una sesión RTP es abierta, al mismo tiempo se abre una sesión RTCP implícita. La función de RTP es proporcionar un medio uniforme de transmisión de datos sometidos a limitaciones de tiempo real (audio, vídeo, etc.).

La función principal de RTCP es informar de la calidad de servicio proporcionada por RTP. Este protocolo recoge estadísticas de la conexión y también información, como por ejemplo bytes enviados, paquetes enviados, paquetes perdidos o Jitter entre otros. Una aplicación puede usar esta información para incrementar la calidad de servicio (QoS), ya sea limitando el flujo o usando un códec de compresión más baja. En resumen. RTCP se usa para informar de la QoS.

## 2.2 Funcionamiento de Wi-Fi

Wi-Fi es un mecanismo de conexión entre dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi pueden conectarse a internet a través de un punto de acceso de red inalámbrica.

Wi-Fi es una marca de la Wi-Fi Alliance, la organización comercial que adopta, prueba y certifica que los equipos cumplen las normas IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) 802.11 relacionados a redes inalámbricas de área local.

Existen diversos tipos de Wi-Fi, y cada uno de ellos se basa en un estándar IEEE 802.11 aprobado, y funcionan en las bandas de 2,4 y 5 GHz (dependiendo de la versión).

Como se comentó en el capítulo anterior, existen otras tecnologías inalámbricas como Bluetooth que también funcionan a una frecuencia de 2,4 GHz, por lo que puede presentar interferencias con la tecnología Wi-Fi.

Existen varios dispositivos Wi-Fi, los cuales se pueden dividir en dos grupos: dispositivos de distribución o de red, entre los que destacan los enrutadores, puntos de acceso y repetidores; y dispositivos terminales que en general son las tarjetas receptoras para conectar al PC, ya sean internas PCI/PCIe o bien USB.

Las redes Wi-Fi poseen una serie de ventajas, entre las cuales podemos destacar:

- Al ser redes inalámbricas, la comodidad que ofrecen es muy superior a las redes cableadas porque cualquiera que tenga acceso a la red puede conectarse desde distintos puntos dentro de un espacio lo bastante amplio.
- Una vez configuradas, las redes Wi-Fi permiten el acceso de múltiples dispositivos sin necesidad de efectuar ningún gasto adicional en infraestructura.
- La Wi-Fi Alliance asegura que la compatibilidad entre dispositivos con la marca Wi-Fi es total, con lo que en cualquier parte del mundo es posible utilizar la tecnología Wi-Fi con una compatibilidad absoluta.

Pero la tecnología Wi-Fi presenta los problemas intrínsecos de cualquier tecnología inalámbrica. Algunos de ellos son:

- Una de las desventajas que tiene el sistema Wi-Fi es una menor velocidad en comparación a una conexión cableada, debido a las interferencias y pérdidas de señal que el ambiente puede acarrear. La potencia de una conexión Wi-Fi se puede también ver afectada por los agentes físicos que se encuentren en la trayectoria entre transmisor y receptor. Esto puede tener una importante repercusión sobre el aplicaciones en tiempo real, como el caso de los clientes de VoIP.
- La desventaja fundamental de estas redes reside en el campo de la seguridad. Existen algunos programas capaces de capturar paquetes, trabajando con su tarjeta Wi-Fi en modo promiscuo, de forma que puedan calcular la contraseña de la red y de esta forma acceder a ella. Las claves de tipo WEP son relativamente fáciles de conseguir con este sistema. La Wi-Fi Alliance arregló estos problemas lanzando el estándar WPA y posteriormente WPA2, basados en el grupo de trabajo IEEE 802.11i (para más información, consultar referencia). [5] Las redes protegidas con WPA2 se consideran robustas dado que proporcionan muy buena seguridad.

Esta tecnología no es compatible con otros tipos de conexiones inalámbricas como Bluetooth, GPRS, UMTS, etc.

### 2.2.1 Estándares 802.11

Los estándares 802.11 comprenden un conjunto de normas del IEEE que definen las capas físicas y de acceso al medio de una red inalámbrica que usa las bandas de 2.4 ó 5.2 GHz para proporcionar tasas de 11 a 540 Mb/s.

La versión original del estándar 802.11, publicada en 1997, especifica dos velocidades de transmisión teóricas de 1 y 2 Mb/s que se transmiten por señales infrarrojas (IR). IR sigue siendo parte del estándar, aunque no hay implementaciones disponibles.

La capa física (a veces abreviada capa "PHY") ofrece tres tipos de codificación de información. La capa de enlace de datos está compuesta por dos subcapas: control de enlace lógico (LLC/IEEE 802.2) [12] y control de

acceso al medio (MAC). En la figura 2-3 puede apreciarse la estructura de las diferentes capas.

La capa física define la modulación de las ondas de radio y las características de señalización para la transmisión de datos mientras que la capa de enlace de datos define la interfaz entre el bus del equipo y la capa física, en particular un método de acceso, CSMA/CA, parecido al utilizado en el estándar Ethernet (CSMA/CD), y las reglas para la comunicación entre las estaciones de la red. En realidad, el estándar 802.11 tiene tres capas físicas que establecen modos de transmisión alternativos: Espectro Ensanchado por Secuencia Directa (DSSS), Espectro Ensanchado por Salto de Frecuencia (FHSS) e Infrarrojo como ya se ha comentado.

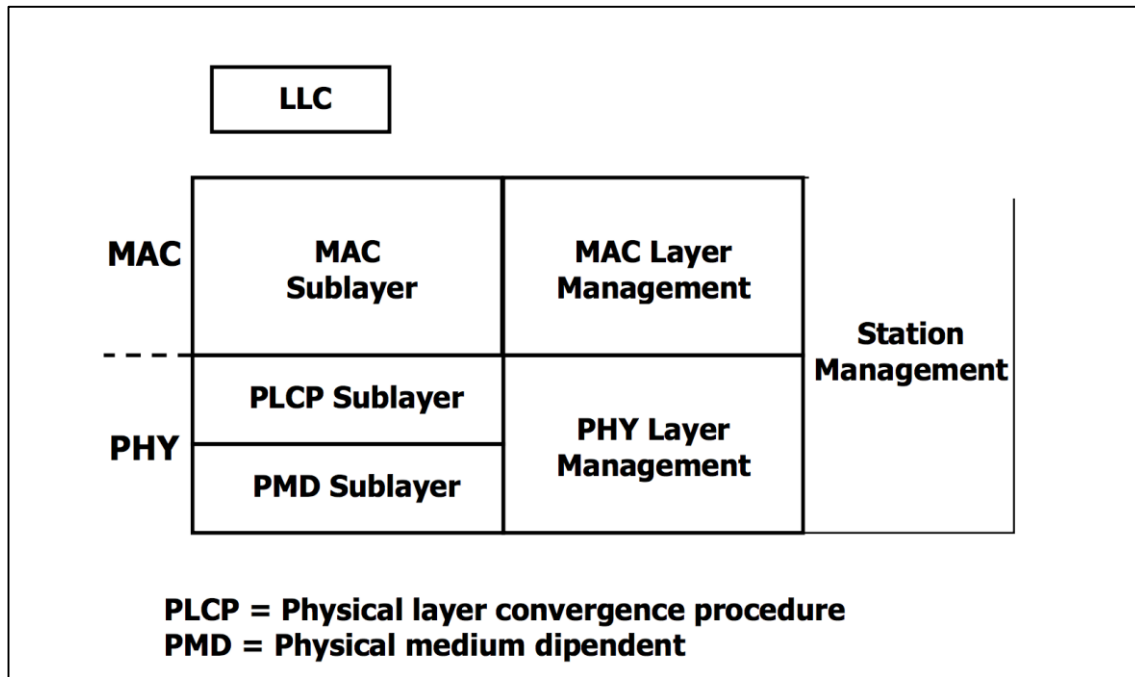


Figura 2-3. Arquitecturas de las capas OSI 1 y 2 del estándar 802.11 [8]

Cualquier protocolo de nivel superior puede utilizarse en una red inalámbrica 802.11 de la misma manera que puede utilizarse en una red Ethernet.

### 2.2.1.1 Arquitectura básica

Los nodos se subdividen en dos tipos: estaciones y puntos de acceso. Las estaciones (STA) corresponden a los terminales móviles de usuario, mientras que los puntos de acceso (APs) son aquellos que actúan de intermediarios entre las estaciones y la red (internet o local).

La arquitectura IEEE 802.11 implica varios componentes que interactúan para proporcionar una WLAN que soporte la movilidad de los nodos de forma transparente para las capas superiores.

El bloque básico de esta arquitectura es lo que se conoce como Basic Service Set (BSS), y está constituido por un punto de acceso y por una o varias estaciones que deben estar asociadas al mismo. Dentro de un BSS el AP se encarga del reenvío de los paquetes de las estaciones, lo cual duplica el consumo de ancho de banda, y permite varias funcionalidades entre las que destaca el modo de ahorro de energía o Power Saving Mode (PSM) que posteriormente desarrollaremos.

También es posible la comunicación directa entre estaciones con topologías Ad-hoc, siendo el bloque básico compuesto por varias estaciones un IBSS.

La interconexión entre los APs se realiza mediante un Distribution System (DS) que suele ser una red cableada, por ejemplo basada en Ethernet, estándar que se tomó como base para la redacción del estándar IEEE 802.3. [13]

No obstante es necesario matizar, que en topologías de red más complejas, los APs se pueden conectar a un controlador que se encarga de gestionar los recursos radio, el traspaso de las estaciones entre APs y el reenvío del tráfico.

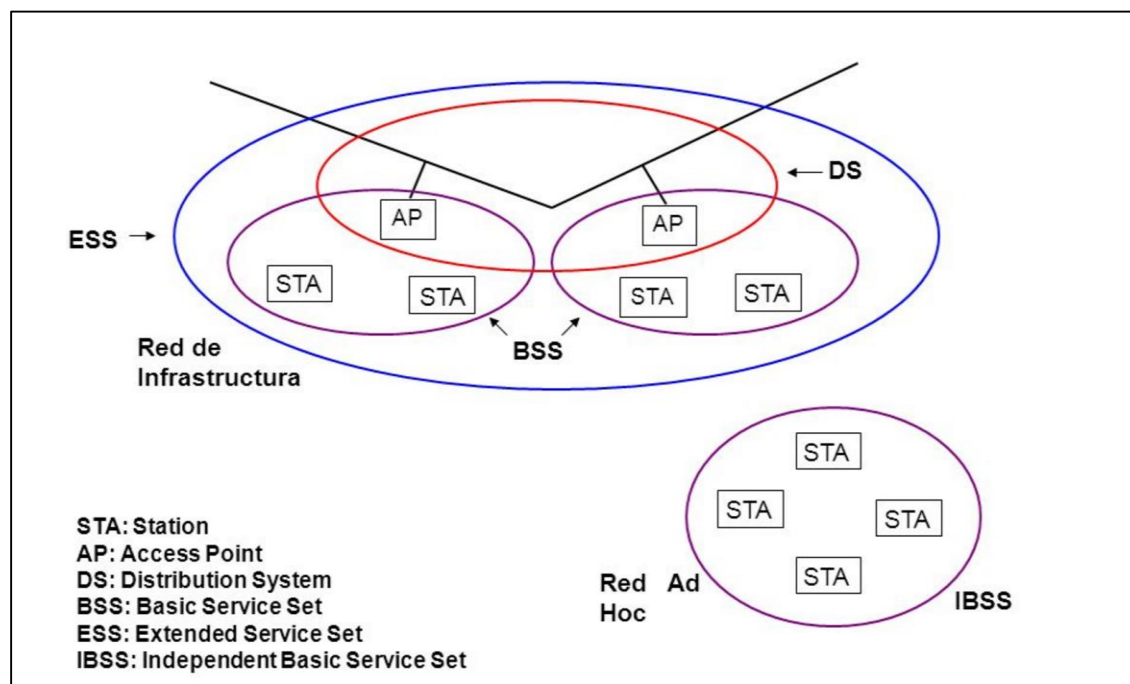


Figura 2-4. Arquitectura de una red 802.11 [9]

### 2.2.1.2 Capa física

La capa física (PHY) es la interfaz entre la capa MAC y el medio inalámbrico. Provee de tres niveles de funcionalidad: intercambio de tramas entre PHY y MAC, generación la portadora de señal y de la modulación de espectro ensanchado para transmitir tramas a través del medio y proveer al nivel MAC de un indicador de detección de portadora para señalar actividad en el medio.

El nivel físico está dividido en dos sub-capas, el PLCP (Physical Layer Convergence Procedure) más cercano a la capa MAC y el PMD (Physical Medium Dependent Layer), que interacciona con el medio inalámbrico. La relación entre ambas se mostró previamente en la Figura 2-3.

Tras la publicación del estándar 802.11 original se han ido realizando revisiones del mismo, que han implicado cambios en la capa física. De esta forma debemos contemplar distintas capas PHY, que han ido evolucionando de 11 a 540 Mb/s: 802.11, 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac. Cada capa física implica un formato de trama física diferente en la mayoría de los casos, como puede observarse a continuación. [13]

#### 2.2.1.2.1 802.11b

La revisión 802.11b del estándar original fue ratificada en 1999.

802.11b tiene una velocidad máxima de transmisión de 11 Mbps y utiliza el mismo método de acceso definido en el estándar original CSMA/CA. El estándar 802.11b funciona en la banda de 2,4 GHz.

Las modulaciones empleadas son DSSS y Modulación por desplazamiento de fase (QPSK) con un ancho de banda aproximado de 22 MHz por canal, pero se emplean 14 canales con 5 MHz de separación entre portadoras, lo que puede generar que estos se solapen. Esto implica una serie de restricciones en la planificación de los puntos de acceso radio, para evitar el solapado de los canales.

Debido al espacio ocupado por la codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 11 Mb/s aunque existen velocidades menores (1, 2, 5.5) cada una de ellas con distinta modulación.

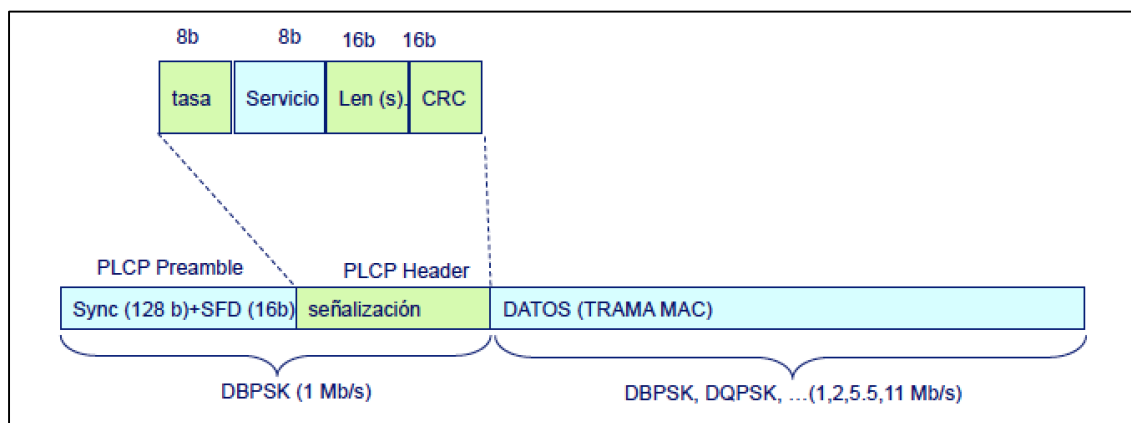


Figura 2-5. Formato de trama física en 802.11b [5]

#### 2.2.1.2.2 802.11a

La revisión 802.11a fue aprobada en 1999. Este estándar utiliza el mismo juego de protocolos de base que el estándar original, aunque opera en la banda de 5 GHz y utiliza 52 subportadoras (48 subportadoras y 4 pilotos) de acceso múltiple por división de frecuencias ortogonales (OFDM) con una velocidad máxima de 54 Mbit/s, lo que lo hace un estándar práctico para redes inalámbricas con velocidades reales de aproximadamente 20 Mbit/s.

Emplea modulaciones por desplazamiento de fase (BPSK, QPSK) y modulaciones de amplitud en cuadratura (16QAM, 64QAM). La velocidad de datos se reduce a 48, 36, 24, 18, 12, 9 o 6 Mbit/s en caso necesario variando el tipo de modulación.

802.11a tiene 12 canales de 20 MHz sin solape, 8 para red inalámbrica y 4 para conexiones punto a punto.

No puede interoperar con equipos del estándar 802.11b, excepto si se dispone de equipos que implementen ambos estándares.

El empleo de la banda de 5GHz implica una serie de consideraciones: se trata de una banda con menor cantidad de interferencias que la banda de 2.4 GHz, aunque el alcance es menor por el aumento de la frecuencia.

#### 2.2.1.2.3 802.11g

En junio de 2003, se ratificó un tercer estándar de modulación: 802.11g, que es la evolución de 802.11b. Este utiliza la banda de 2,4 GHz (al igual que 802.11b) pero opera a una velocidad teórica máxima de 54 Mbit/s, que en promedio es de 22,0 Mbit/s de velocidad real de transferencia, similar a la del estándar 802.11a. Es compatible con el estándar b y utiliza las mismas frecuencias y retrocede también a las modulaciones DSSS, QPSK por preservar la compatibilidad.

Buena parte del proceso de diseño del nuevo estándar lo tomó el hacer compatibles ambos modelos. Sin embargo, en redes bajo el estándar g la presencia de nodos bajo el estándar b reduce significativamente la velocidad de transmisión.

Los equipos que trabajan bajo el estándar 802.11g llegaron al mercado muy rápidamente, y esto se debió en parte a que para construir equipos bajo este nuevo estándar se podían adaptar los ya diseñados para el estándar b.

Actualmente se venden equipos con esta especificación, con potencias de hasta medio vatio, que permite realizar comunicaciones de más de 50 km con antenas parabólicas o equipos de radio apropiados.

Existe una variante llamada 802.11g+ capaz de alcanzar los 108Mbps de tasa de transferencia. Generalmente solo funciona en equipos del mismo fabricante ya que utiliza protocolos propietarios.

#### 2.2.1.2.4 802.11n

El estándar 802.11n fue ratificado por la organización IEEE el 11 de septiembre de 2009 con una velocidad de 600 Mbps en capa física, haciendo uso simultáneo de las bandas de frecuencia de 2.4 GHz y 5 GHz, con un



ancho de banda por canal de 20 MHz.

Está construido basándose en estándares previos de la familia 802.11, agregando Multiple-Input Multiple-Output (MIMO) y unión de interfaces de red (Channel Bonding).

MIMO es una tecnología que emplea múltiples antenas transmisoras y receptoras (hasta cuatro) para mejorar el desempeño del sistema, permitiendo manejar más información que al utilizar una sola antena. Dos beneficios importantes que provee a 802.11n, son la diversidad de antenas y el multiplexado espacial.

Channel Bonding, también conocido como 40 MHz o unión de interfaces de red, es la segunda tecnología incorporada al estándar 802.11n la cual permite utilizar dos canales separados de 20 MHz, que no se solapen, para transmitir datos simultáneamente. Esto permite doblar la velocidad de la capa física disponible en un solo canal de 20 MHz.

Utilizar conjuntamente una arquitectura MIMO con canales de mayor ancho de banda, ofrece la oportunidad de crear sistemas muy poderosos y rentables para incrementar la velocidad de transmisión de la capa física.

Se conoce que el futuro estándar sustituto de 802.11n será 802.11ac con tasas de transferencia superiores a 1 Gb/s.

#### 2.2.1.2.5 802.11ac

802.11ac, también conocido como Wi-Fi 5G o Wi-Fi Gigabit, es una mejora de la norma 802.11n desarrollada entre el año 2011 y el 2013, y finalmente aprobada en enero de 2014.

El estándar se centra en mejorar las tasas de transferencia hasta 433 Mbit/s por flujo de datos, consiguiendo teóricamente tasas de 1.3 Gbit/s empleando 3 antenas. Opera dentro de la banda de 5 GHz, amplía el ancho de banda hasta 160 MHz, con canales simples de 80 MHz y utiliza hasta 8 flujos MIMO.

Este estándar incluye modulación de amplitud en cuadratura de alta densidad (256 QAM), que incrementa la eficiencia en la transferencia de datos entre dispositivos reduciendo el consumo de batería en consecuencia.

Debido a que se utilizará la banda de los 5GHz, en la que como hemos comentado, las antenas tienen menos alcance que en la banda de 2.4GHz en las mismas condiciones, este nuevo estándar incluye Beamforming para transmisión y recepción.

El Beamforming es un tipo de categoría MIMO que consiste en la formación de una onda de señal reforzada mediante el desfase en distintas antenas y es capaz de “superar” obstáculos llegando hasta al cliente por el mejor camino. El Beamforming reconoce los elementos que causan un bajo rendimiento (muros, paredes) e intenta evitarlos.

Las tecnologías LPDC, (Low density parity check) y STBC (Space-Time Block Codes) también están presentes en este nuevo estándar para aumentar la eficiencia de la transmisión al máximo y conseguir altas velocidades inalámbricas. [11]

#### 2.2.1.3 Capa MAC original

El estándar 802.11 define en su capa de control de acceso al medio una serie de funciones para realizar las operaciones propias de las redes inalámbricas. La capa MAC se encarga, en general, de gestionar y mantener las comunicaciones entre estaciones 802.11, bien sean puntos de acceso o adaptadores de red. Por tanto, tiene que coordinar el acceso a un canal de radio compartido y utilizar su capa física para detectar la portadora, y transmitir y recibir tramas.

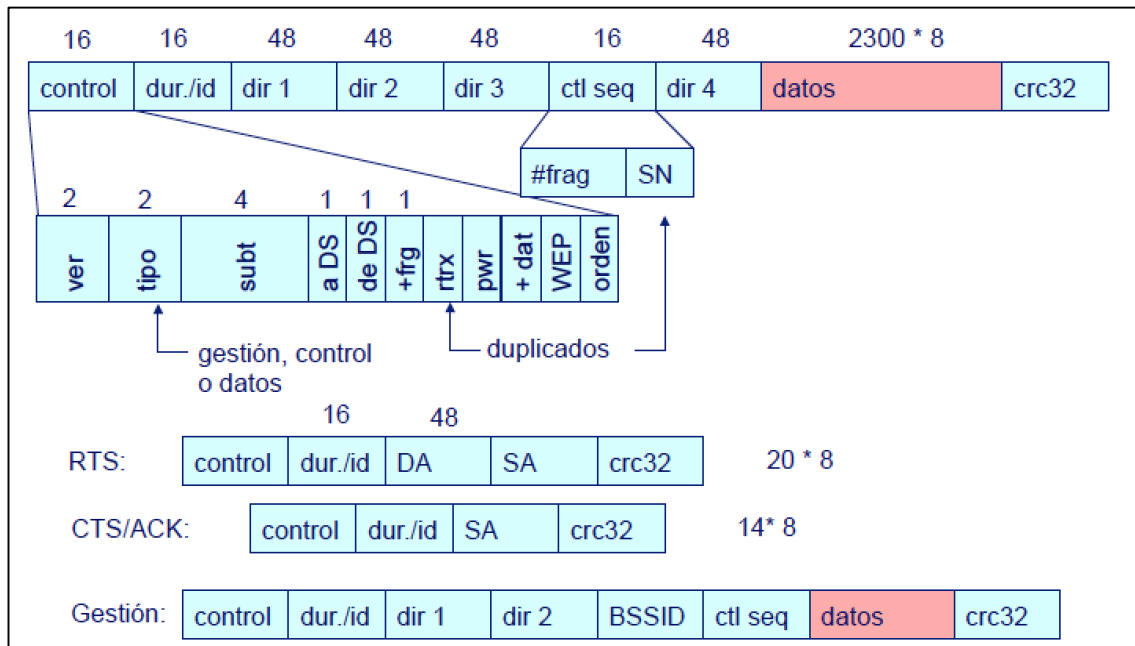


Figura 2-6. Formatos de trama MAC en 802.11 [5]

Entre las principales funcionalidades de la capa MAC se encuentran la búsqueda tanto activa como pasiva de APs, procedimientos de autenticación para comprobar la identidad de una determinada estación, procedimientos de asociación por los cuales el AP reserva recursos y se sincroniza con una estación cliente y fragmentación de tramas.

Una estación tiene que obtener primero el acceso al medio antes de poder transmitir tramas. El medio, como se ha comentado ya, es un canal de radio compartido. El estándar 802.11 define dos formas de acceso al medio, función de coordinación distribuida (DCF) y función de coordinación de punto (PCF), aunque esta última no goza de una gran popularidad a nivel comercial.

DCF es obligatorio en todas las estaciones inalámbricas y se emplea el método CSMA/CA con un algoritmo binario exponencial denominado backoff. Una estación sólo puede transmitir cuando el canal está libre, si otra estación envía una trama debe esperar a que el canal esté libre para poder transmitir. [12] [10]

#### 2.2.1.3.1 DCF y CSMA/CA

CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) o, en español, acceso múltiple con escucha de portadora y evasión de colisiones, es un protocolo de control de acceso a redes de bajo nivel que permite que múltiples estaciones utilicen un mismo medio de transmisión. Cada equipo anuncia opcionalmente su intención de transmitir antes de hacerlo para evitar colisiones entre los paquetes de datos. De esta forma, el resto de equipos de la red sabrán cuando hay colisiones y en lugar de transmitir la trama en cuanto el medio está libre, se espera un tiempo aleatorio adicional corto y solamente si, tras ese corto intervalo el medio sigue libre, se procede a la transmisión reduciendo la probabilidad de colisiones en el canal.

Para poder entender los esquemas que se muestran a continuación, es necesario comentar alguno de los parámetros que intervienen en los mismos (el valor de los intervalos varía en función de la capa física empleada) [13]:

- SIFS (short inter frame space): se emplea cuando una estación necesita mantener el medio para transmisiones adicionales como un ACK o CTS/RTS (se explican a continuación) sin que otras estaciones interfieran.
- DIFS (DCF inter frame space): se emplea como intervalo de espera tras las tramas de datos ordinarias.
- PIFS (PCF inter frame space): Se calcula como la suma de SIFS y la duración de un Time Slot. La duración de PIFS es menor que DIFS y mayor que SIFS.
- EIFS (extended inter frame space): se emplea como intervalo de espera tras la detección de una trama errónea.

- Time slot ( $T_s$ ), es un intervalo de tiempo (ranura) que se emplea como unidad de medida para el proceso de backoff.
- Contention window (ventana de contención). El tamaño de la ventana de contención se define por el número de slots de la misma.
- ACK(acknowledgement): Tras recibir una trama, cada estación realiza un proceso de comprobación de errores empleando el CRC de la trama. Si no hay errores se envía un ACK a la estación emisora de la trama. Si la estación emisora no recibe el ACK dentro de un determinado periodo de tiempo, se retransmite la trama.

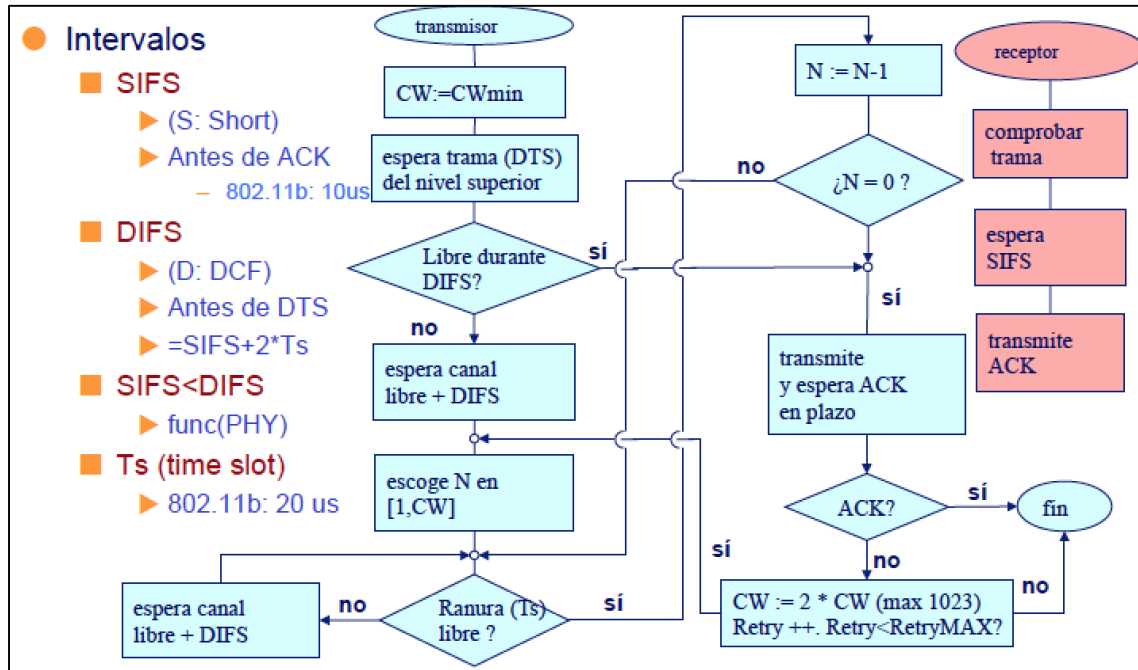


Figura 2-7. Esquema simplificado de CSMA/CA [5]

Cuando la capa MAC recibe una solicitud de transmisión, se sondea el medio para ver si está ocupado. Si está libre durante un periodo igual a DIFS o EIFS, la transmisión puede comenzar.

Si el medio se encontrase ocupado, el nivel MAC elegiría un periodo de backoff aleatorio, en un rango comprendido entre 1 y el tamaño de la ventana de contención empleando un algoritmo. El periodo de backoff es la cantidad de tiempo durante el que la estación debe escuchar el medio libre antes de intentar transmitir nuevamente. Con cada intento de transmisión fallido el umbral superior del rango se duplica hasta que se alcanza un valor máximo de intentos. Si se consigue transmitir correctamente una trama, el valor de la ventana se reduce hasta su mínimo.

El valor del backoff se decrementa con cada Time Slot durante el que el medio se ha encontrado libre, y cuando el valor del backoff llega a 0 la estación puede comenzar la transmisión. Es posible que dos estaciones tomen el mismo intervalo de backoff, lo que generará una colisión.

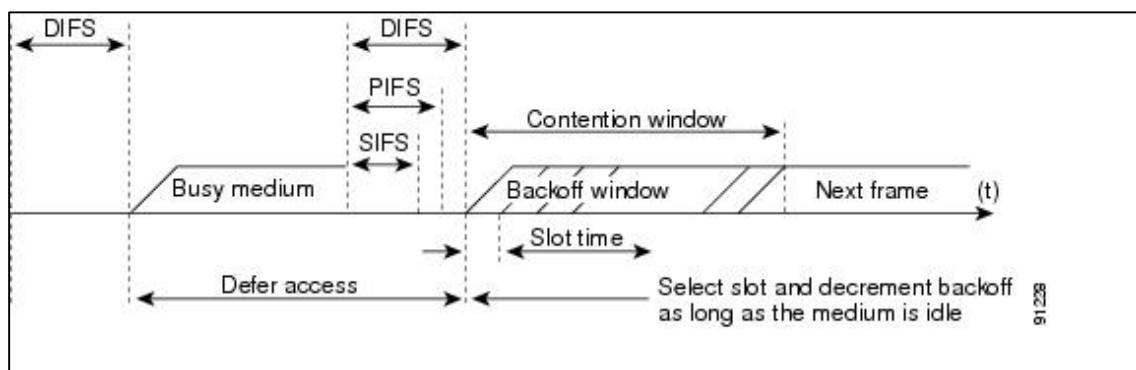


Figura 2-8. Acceso básico en DCF empleando CSMA/CA [6]

Aunque CSMA/CA aseguren que un nodo va a obtener un acceso al medio no, se asegura que el nodo destino esté en contacto con el nodo origen, lo que se conoce como el problema del “nodo oculto”. Para solucionar este problema se añade un procedimiento de saludo adicional al protocolo de la capa MAC.

Para enviar una trama, el equipo origen primero envía una trama corta de control de solicitud de transmisión RTS (Request To Send) mediante el método CSMA/CD o CSMA/CA. Este mensaje de control RTS contiene las direcciones de MAC del equipo origen y destino.

Si el equipo destino recibe esta trama, significa que está preparado para recibir una trama. Este equipo devolverá una trama de contestación: preparado para transmitir CTS (Clear To Send) o receptor ocupado (RxBUSY). Si la respuesta es afirmativa, el equipo origen transmite la trama en espera (DATA). Si el equipo destino recibe correctamente el mensaje, contesta con la trama de confirmación positiva ACK y si no la recibe correctamente, contesta con la trama de confirmación negativa NAK y el equipo origen tratará de volver a enviarlo. Este procedimiento se repite un número predefinido de veces hasta conseguirse una transmisión correcta de la trama DATA.

Este proceso de solicitud de envío evita colisiones entre nodos ocultos. El saludo RTS/CTS continúa en cada trama mientras que el tamaño de la trama exceda del umbral establecido en el adaptador correspondiente. En la mayoría de adaptadores de red los usuarios pueden fijar un umbral máximo de tamaño de trama para que el adaptador de red active RTS/CTS. Por ejemplo, si establecemos un tamaño de trama de 1.000 bytes, cualquier trama de un tamaño superior a 1.000 bytes disparará RTS/CTS. De esta forma el proceso sólo afectaría a las tramas más grandes y más costosas de retransmitir pero las más pequeñas es mejor arriesgarse. [13]

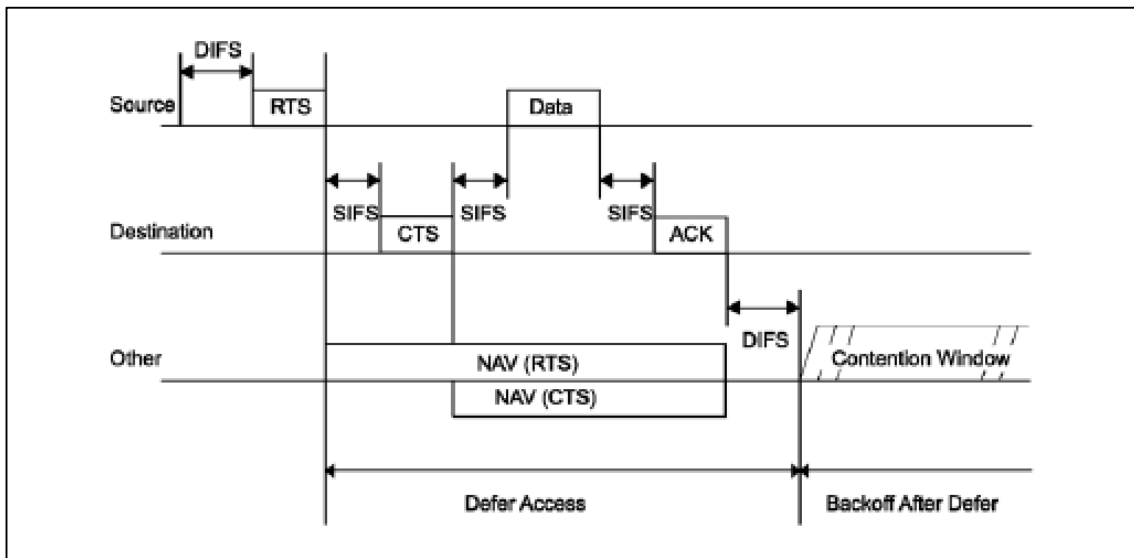


Figura 2-9. Envío con reserva previa [6]

En la siguiente figura puede apreciarse el funcionamiento del mecanismo con varias estaciones compitiendo por acceder al canal. El tiempo de transmisión de trama incluye el intercambio RTS/CTS y ACK. CP simboliza el periodo de contención durante el que se realiza el proceso de backoff.

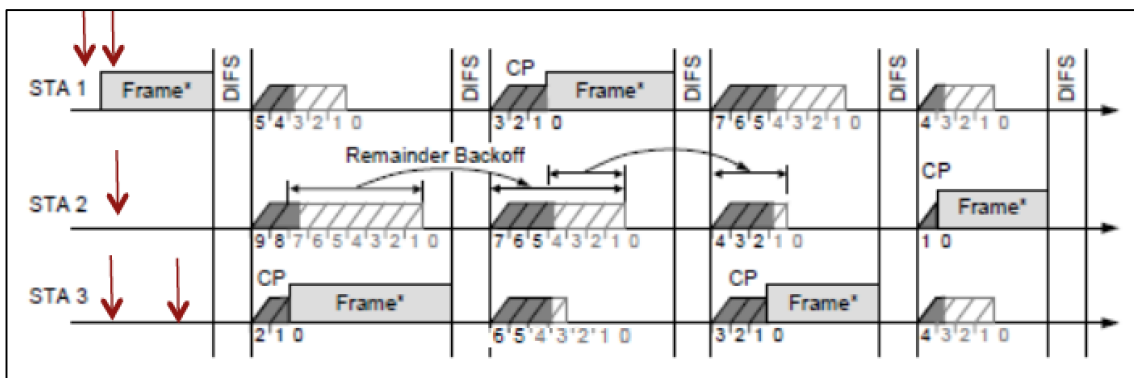


Figura 2-10. Mecanismo de Backoff con varias estaciones [5]

### 2.2.1.3.2 Power Saving Mode (802.11 legacy)

El funcionamiento normal de las redes inalámbricas supone un acceso constante al medio es decir, se escucha de forma constante el medio con el consiguiente consumo de energía. 802.11 establece unos mecanismos para intentar reducirlo. El mecanismo de Power Saving Mode consiste en apagar (realmente llevarlo a un estado de muy bajo consumo) el adaptador de una estación y hacer que se active en periodos regulares en busca de una trama Beacon o baliza, un tipo concreto de las tramas de gestión del nivel MAC de 802.11 que contiene el TIM (Traffic Information Map) [13]. Durante el tiempo que transcurre entre paquetes con TIM, el adaptador se desactiva para ahorrar energía. Todos los adaptadores de una red tienen que activarse simultáneamente para escuchar el TIM del punto de acceso.

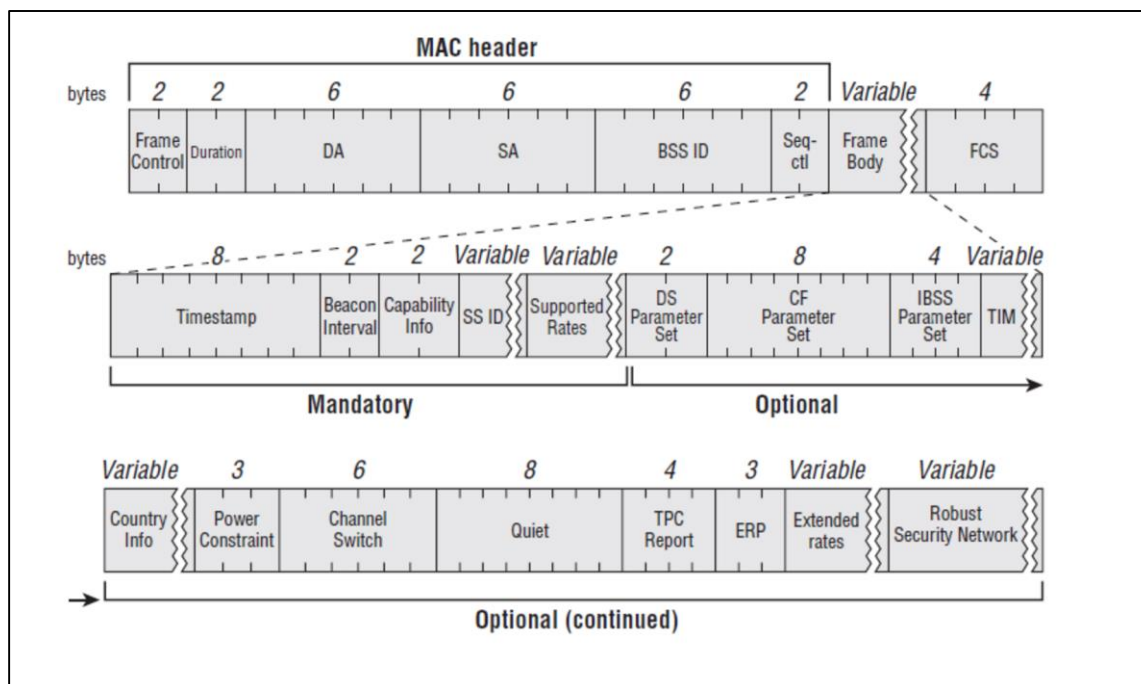


Figura 2-11. Formato genérico de una trama Beacon [7]

El TIM informa a los clientes que tienen datos pendientes en el punto de acceso. Para ello se emplea un mapa de bits en el que cada uno de ellos representa el Identificador de Asociación (AID) de la estación, puesto a 1 si tiene 1 o más tramas pendientes. Cuando un adaptador sabe mediante el TIM que tiene datos pendientes, envía una trama PS-Poll solicitando los datos almacenados y permanece activo el tiempo necesario para recibirlos. El punto de acceso dispone de un buffer para almacenar los datos hasta que los envía a la estación. El AP almacena las tramas para las estaciones durante al menos el tiempo de sueño que indicaron en la asociación. Una vez que la estación ha recibido sus datos, entonces vuelve al modo inactivo.

Un punto de acceso indica la presencia de tráfico de difusión mediante paquetes DTIM (delivery traffic information map). DTIM es un temporizador múltiplo de TIM. Gracias a este valor, que podemos configurar en el punto de acceso, podemos especificar cuanto tiempo tiene que permanecer una estación activa para buscar tráfico de difusión, y por tanto cuanto más largo sea el intervalo DTIM, mayor será el tiempo en el que la estación puede dormir y ahorrar energía. [12] [13]

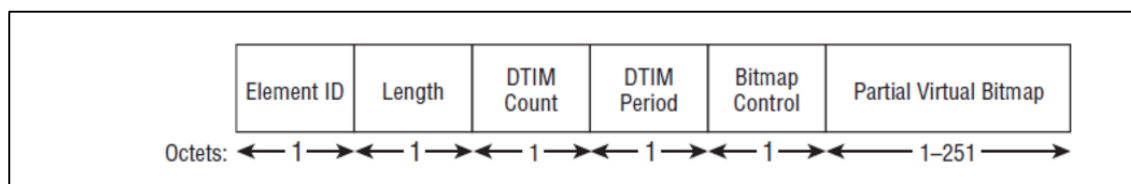


Figura 2-12. Formato del TIM [6]

Los dos primeros campos del TIM indican que se trata de un elemento de tipo TIM (5) y la longitud total del mismo.

DTIM count cuenta de forma incremental el número de tramas beacon que se han ido enviando hasta el siguiente DTIM, mientras que DTIM period indica el número de tramas beacon que se envían entre dos tramas

## DTIM.

El campo Bitmap Control indica si las tramas multicast o unicast son almacenadas en el AP y se emplea como offset del mapa de bits

El último campo es el que contiene el mapa de bits que representa los AIDs de las estaciones e indica si tienen tramas pendientes de recibir.

La estación solicita el envío de las tramas almacenadas en el AP mediante el empleo de tramas PS-Poll, y el AP le responde enviando una trama almacenada. Con el bit More Data se indica si quedan otras tramas pendientes de envío.

El formato paquete es el mismo para TIM y DTIM.

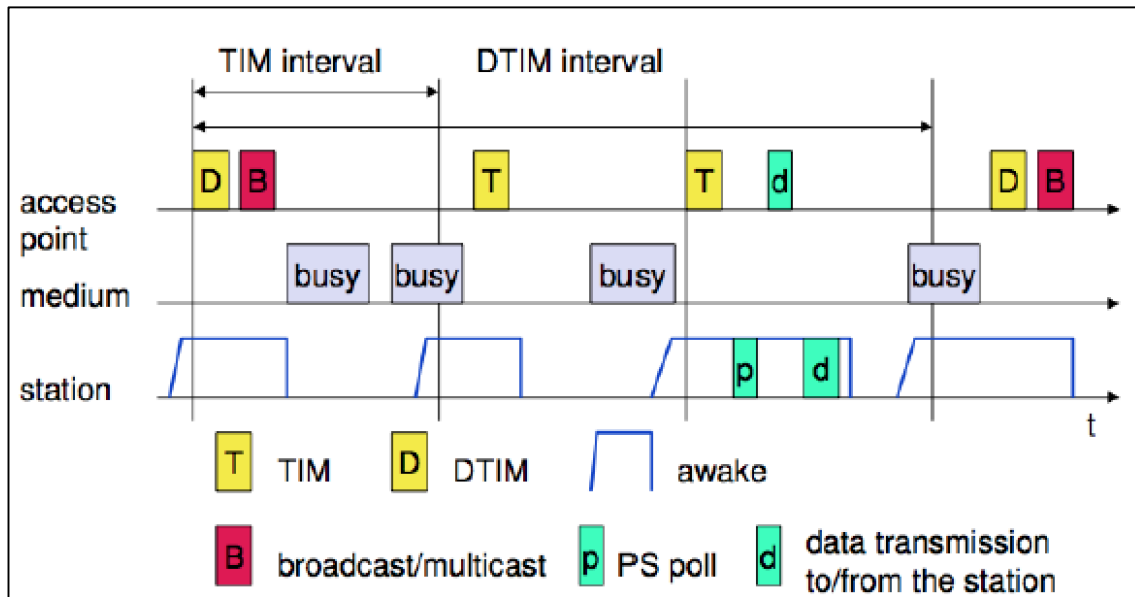


Figura 2-12. Intercambio de tramas en PSM [5]

Debido a la duración de los intervalos de Beacon y al empleo de señalización adicional para conocer el estado del buffer y descargar los paquetes, este mecanismo no es recomendable para aplicaciones en tiempo real con requisitos específicos de retardo y pérdidas como VoIP.

### 2.2.1.4 802.11e y WMM

802.11e mejora el DCF y PCF, a través de una nueva función de coordinación: la función de coordinación híbrida (HCF). Dentro de esta, hay dos métodos de acceso al canal, similares a los definidos en 802.11: HCCA (acceso controlado al canal), muy parecido a PCF, y EDCA (acceso distribuido al canal mejorado). Tanto EDCA y HCCA definen categorías de tráfico (TC).

En EDCA, el tráfico de alta prioridad tiene una mayor probabilidad de ser enviado que el tráfico de baja prioridad: una estación con tráfico de alta prioridad espera un poco menos antes de enviar su paquete, en promedio, que una estación con tráfico de baja prioridad. Esto se logra mediante el protocolo TCMA, que es una variación de CSMA / CA utilizando un más corto arbitraje espacio entre tramas (AIFS) para los paquetes de prioridad más alta. Los valores exactos dependen de la capa física que se utiliza para transmitir los datos.

Además, EDCA proporciona acceso libre de contienda al canal por un período denominado oportunidad de transmisión (TXOP). TXOP es un intervalo de tiempo limitado durante el cual una estación puede enviar tantas tramas como sea posible, empleando el intervalo SIFS como separación (siempre que la duración de las transmisiones no se extienda más allá de la duración máxima de TXOP). Si una trama es demasiado grande para ser transmitida en un solo TXOP, debe ser fragmentada en tramas más pequeñas. El uso de las TXOP reduce el problema de que las estaciones con tasa bajas ganen el canal durante demasiado tiempo. Un intervalo de tiempo TXOP de 0 significa que se limita a una única trama, y el tiempo que el canal es ocupado lo determina la longitud de la misma y la tasa de transmisión, como en DCF.

EDCA permite que todas las tramas enviadas durante un intervalo TXOP puedan ser asentidas con una única



trama de ACK (Block-ACK), lo cual reduce la sobrecarga de protocolo.

EDCA permite así mismo asignar dos valores específicos para QoS que determinarán los paquetes que deben ser asentidos: QoSAck y QoSNoAck. Esto evita la retransmisión de paquetes para los que el tiempo es crítico.

Los niveles de prioridad en EDCA se denominan categorías de acceso (ACS). La ventana de contención (CW) se puede ajustar de acuerdo con el tráfico previsto en cada categoría de acceso, con una ventana más ancha necesaria para las categorías de tráfico pesado. Los valores CWmin y CWmax se calculan a partir aCWmin y aCWmax valores, respectivamente, que se definen para cada capa física con el apoyo de 802.11e.

Cada categoría de acceso se define con 4 parámetros (Cwmin : Cwmax : MaxTXop : AIFSN).

| AC                  | CWmin | CWmax | AIFSN | Max TXOP |
|---------------------|-------|-------|-------|----------|
| Background (AC_BK)  | 15    | 1023  | 7     | 0        |
| Best Effort (AC_BE) | 15    | 1023  | 3     | 0        |
| Video (AC_VI)       | 7     | 15    | 2     | 3.008ms  |
| Voice (AC_VO)       | 3     | 7     | 2     | 1.504ms  |
| Legacy DCF          | 15    | 1023  | 2     | 0        |

Figura 2-13. Parámetros de las categorías de acceso en 802.11e para la capa física 802.11n [8]

A diferencia de DCF, para cada categoría de acceso se ejecuta una instancia de backoff con distintos parámetros. El intervalo AIFS se calcula como la suma de SIFS y el AIFSN de cada categoría de acceso multiplicado por la duración del Time Slot.

Tras cada colisión, la ventana de contención se recalcula de una forma diferente a la vista en DCF:

$$\text{newCW}(\text{TC}) = (\text{oldCW}(\text{TC}) * \text{PF}) - 1$$

hasta alcanzar un valor máximo, y siendo PF el factor de persistencia, que disminuye conforme aumenta la prioridad de la categoría de acceso.

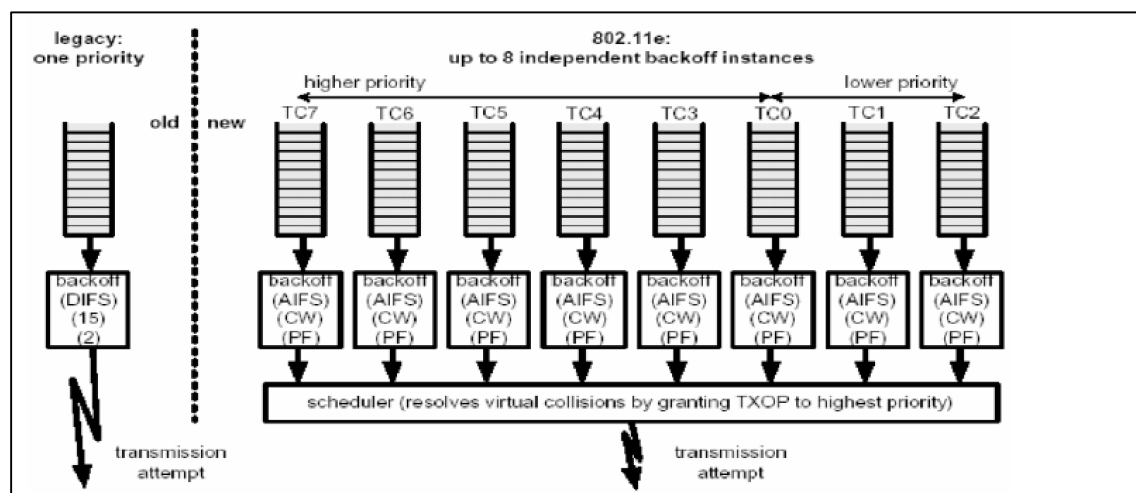


Figura 2-14. Comparación de los parámetros en el estándar original y en 802.11e [5]

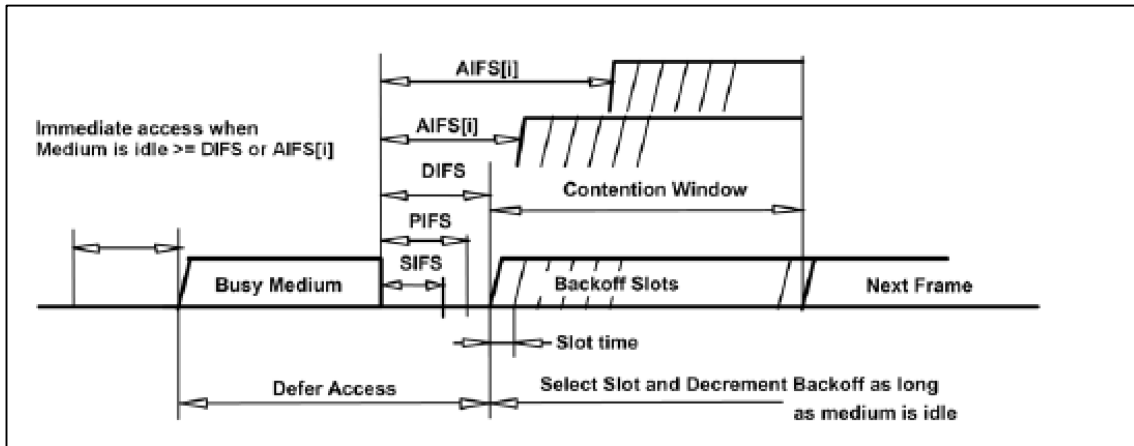


Figura 2-15. Acceso básico en EDCA [9]

La Wi-Fi Alliance define la especificación de Wi-Fi Multimedia (WMM) como un subconjunto del estándar 802.11e. Los APs certificados deben soportar al menos EDCA y TXOP, el resto de mejoras son opcionales.

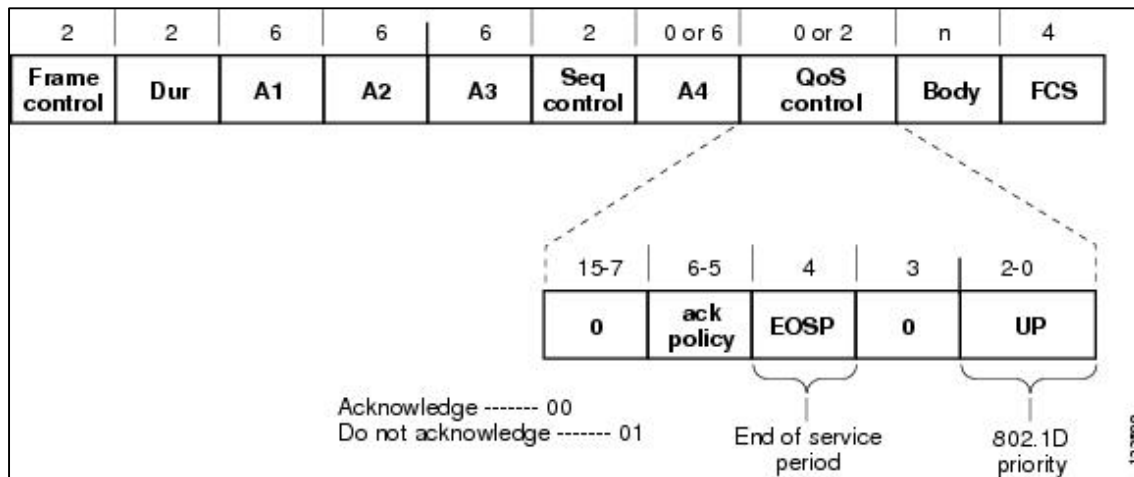


Figura 2-16. Formato de trama MAC en WMM [9]

### 2.2.1.4.1 802.11e APSD

Como se comentó, la modalidad de PSM definida en 802.11 implica retardos no admisibles en algunas aplicaciones sensibles al retardo, como la que nos ocupa en este estudio: VoIP.

802.11e incluye nuevos mecanismos para el ahorro de energía basados en el Automatic Power Save delivery (APSD); programado (S-APSD) y no programado (U-APSD). Con APSD múltiples tramas pueden ser transmitidas juntas por el AP a un dispositivo que esté ejecutando el modo de ahorro de energía durante un periodo de servicio. Al finalizar el periodo de servicio, el dispositivo entra en un estado de “sueño” (doze) hasta que comience el siguiente periodo de servicio.

APSD es un método de administración de energía más eficiente que el definido en el estándar 802.11 original basado en el sondeo (polling). Tiene un menor consumo de energía, ya que reduce tanto el tráfico de señalización que de otro modo sería necesario para la entrega de las tramas almacenadas en el búfer de los APs, así como la tasa de colisiones entre los mensajes de polling, transmitidos normalmente inmediatamente después de la baliza que contiene el TIM. Para una comparación más detallada entre ambos mecanismos, se proporciona la siguiente referencia. [11]

Con S-APSD, los periodos de servicio comienzan de acuerdo a un “programa” (schedule) conocido por los dispositivos, permitiendo así que el AP pueda transmitir el tráfico almacenado sin necesidad de señalización adicional.

U-APSD, mecanismo de ahorro de energía en el que se basa la certificación WMM Power Save (certificación de la Wi-Fi Alliance), permite una transmisión más eficiente y flexible al permitir a cada aplicación controlar los requisitos de capacidad y latencia, decidiendo con qué frecuencia la estación se deberá comunicar con el



AP y cuanto tiempo puede permanecer en estado de sueño.

Las aplicaciones que no emplean U-APSD pueden coexistir con las aplicaciones que si lo hacen dentro de un mismo dispositivo. En este caso, emplearán el procedimiento de ahorro de energía definido en el estándar 802.11 original.

El modo de ahorro de energía concreto que se empleará es negociado por cada estación con el AP durante el procedimiento de asociación. De esta forma, para cada categoría de acceso, se establece un modo concreto. Es importante tener en cuenta que las en las redes Wi-Fi con WMM, el tráfico de voz recibe prioridad sobre los demás tipos de tráfico.

Mientras que las estaciones que emplean PSM del estándar 802.11 original deben esperar a la trama Beacon para iniciar la descarga de los datos almacenados en el AP, los que emplean WMM-PS pueden iniciarla en cualquier momento, es decir, la estación “se despierta” cuando quiere, permitiendo una transmisión de datos más frecuente para aquellas aplicaciones que así lo requieren.

La estación envía una trama “trigger” para cualquiera de las AC que estén empleando WMM-PS para indicar que está despierta y lista para descargar cualquier trama de datos que el AP pueda tener almacenada. A diferencia del PSM original, la trama trigger puede ser cualquier trama de datos, eliminando así la necesidad de emplear tramas PS-Poll para el empleo de señalización. En el caso de no disponer de datos se emplea una trama QoS Null.

Una vez que la estación ha enviado la trama trigger, el AP indica que está listo para enviarle tramas almacenadas. Las tramas de datos se envían durante TXOP, intercalando un ACK de la estación por cada trama correctamente recibida. En la última trama, el AP indica que no hay más tramas pendientes para la estación y que esta puede volver a dormir.

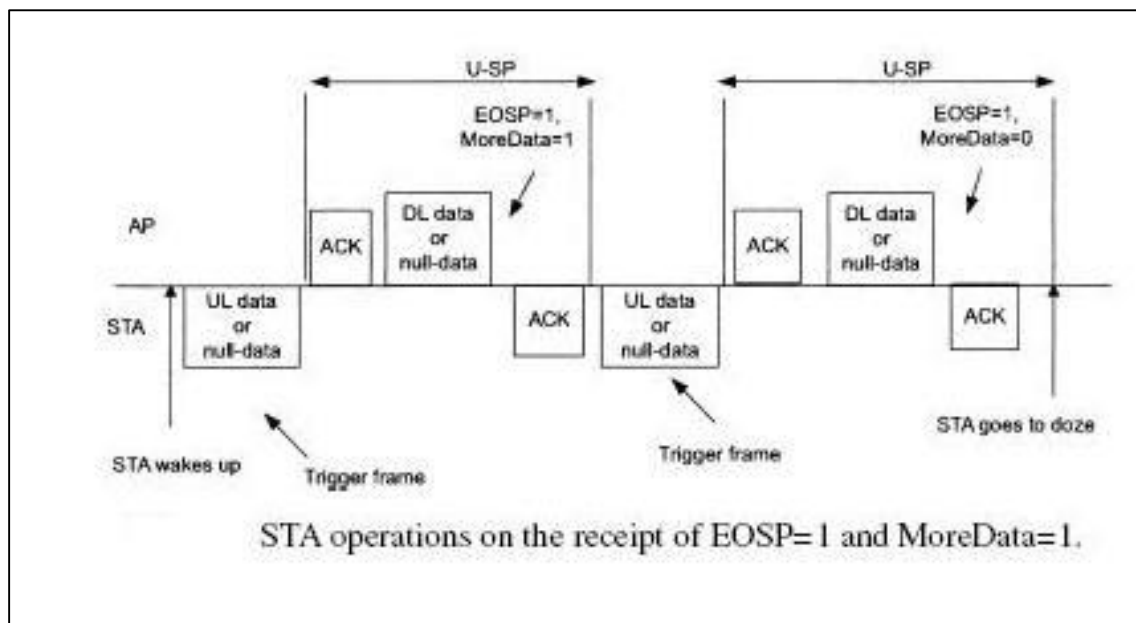


Figura 2-17. Funcionamiento básico de U-APSD [11]

Nótese que EOSP (se muestra en la figura 16) indica el fin del periodo de servicio, y con More Data se indica que la estación tiene tramas pendientes de recibir aún. [8] [12]

A continuación se muestra un esquema más completo en el que intervienen distintas categorías de tráfico dentro de una misma estación.

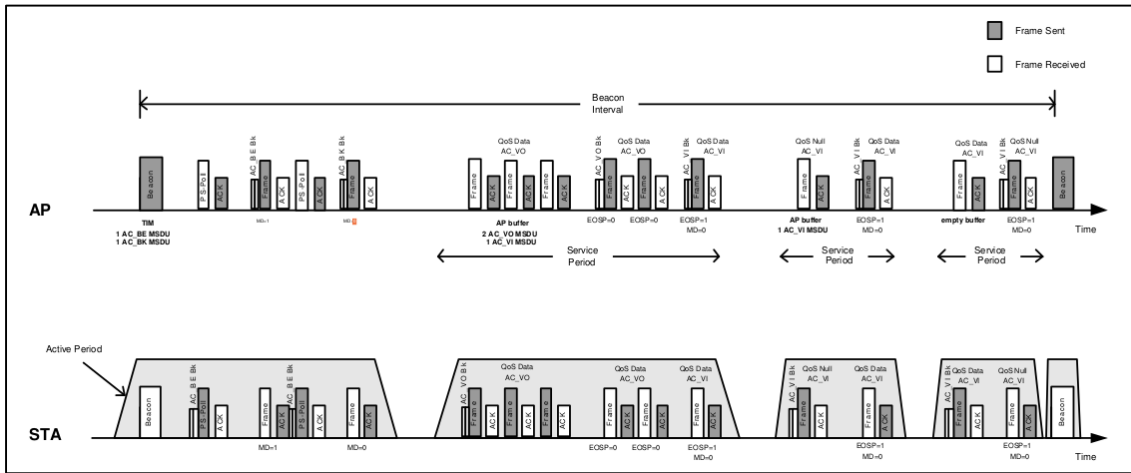


Figura 2-18. Las categorías de acceso AC\_BE y AC\_BK emplean PSM original, mientras que AC\_VI Y AC\_VO implementan U-APSD [14]

Puede observarse como en el primer periodo de actividad de la estación se emplea el mecanismo original de PSM para las categorías previamente señaladas. En el segundo periodo la estación se despierta para enviar un tráfico de voz (AC\_VO), actuando como trigger para iniciar un periodo de servicio no programado la primera trama de voz recibida por el AP. En el tercer periodo, la estación se despierta porque el algoritmo de U-APSD ha decidido que debe iniciarse un periodo de servicio, en este caso la estación envía una trama de QoS null para iniciar el periodo de servicio. Por último, en el cuarto periodo la estación se despierta al tener tráfico de video para enviar (AC\_VI).

# 3 PRUEBAS E INVESTIGACIÓN

---

*If I knew what I was doing, it wouldn't be called research.*

Albert Einstein.

**E**n este apartado se analizan los diferentes dispositivos que se han empleado para el desarrollo de las pruebas, así como su instalación y funcionamiento en diferentes sistemas operativos. Las cuestiones de configuración más avanzadas pueden consultarse en el Anexo A.

## 3.1 Dispositivos Wi-Fi

Para el desarrollo del proyecto se emplearon inicialmente 4 adaptadores Wi-Fi USB y un AP, todos ellos proporcionados por el departamento de telemática de la Escuela Técnica Superior de Ingeniería de Sevilla. Posteriormente, y debido a las necesidades del proyecto, se adquirieron más adaptadores inalámbricos, tanto PCIe como USB.

Con el objeto de ser fiel al desarrollo de las pruebas e investigación se irán presentando los adaptadores en orden de adquisición, y se subdividirá el análisis de las pruebas en dos fases.

### 3.1.1 Edimax EW7811

Se trata de un adaptador USB de reducido tamaño (nanoadaptador) y de altas prestaciones, con soporte para los estándares 802.11b/g/n, tasas superiores a los 150 Mbps, y Green WLAN, una tecnología que según las especificaciones ofrecidas por el fabricante permite ahorrar entre un 20% y un 50% en el consumo energético del adaptador en estado de reposo. No obstante, investigando los documentos que proporciona el fabricante con el driver, puede observarse que este mecanismo consiste simplemente en esperar 3 intervalos de Beacon en lugar de uno al emplear el modo de ahorro de energía tradicional (en la tabla comparativa que puede encontrarse a continuación se proporciona el enlace a los drivers y la documentación).

### 3.1.2 TP-Link TL-WN725N

Este adaptador es muy similar al anteriormente comentado: se trata de un nanoadaptador con dimensiones similares, tasas superiores a los 150 Mbps y cumple igualmente con los estándares 802.11b/g/n. Incorpora un software que permite emplearlo como un “SoftAP” para dar acceso a internet a otros dispositivos inalámbricos.

### 3.1.3 D-Link DWA-160 rev.A2

Este adaptador USB tiene unas dimensiones superiores a las de los anteriormente comentados, y dispone de más funcionalidades, como por ejemplo el soporte de las bandas de frecuencia de 2,4 y 5 GHz así como un botón para el empleo de WPS y tasas de transferencia superiores a 300 Mbps.

### 3.1.4 D-Link DWA-160 rev.C1

Se trata de una revisión más moderna del hardware del modelo anterior, con características similares, con un tamaño un poco más reducido y que conserva el botón WPS.

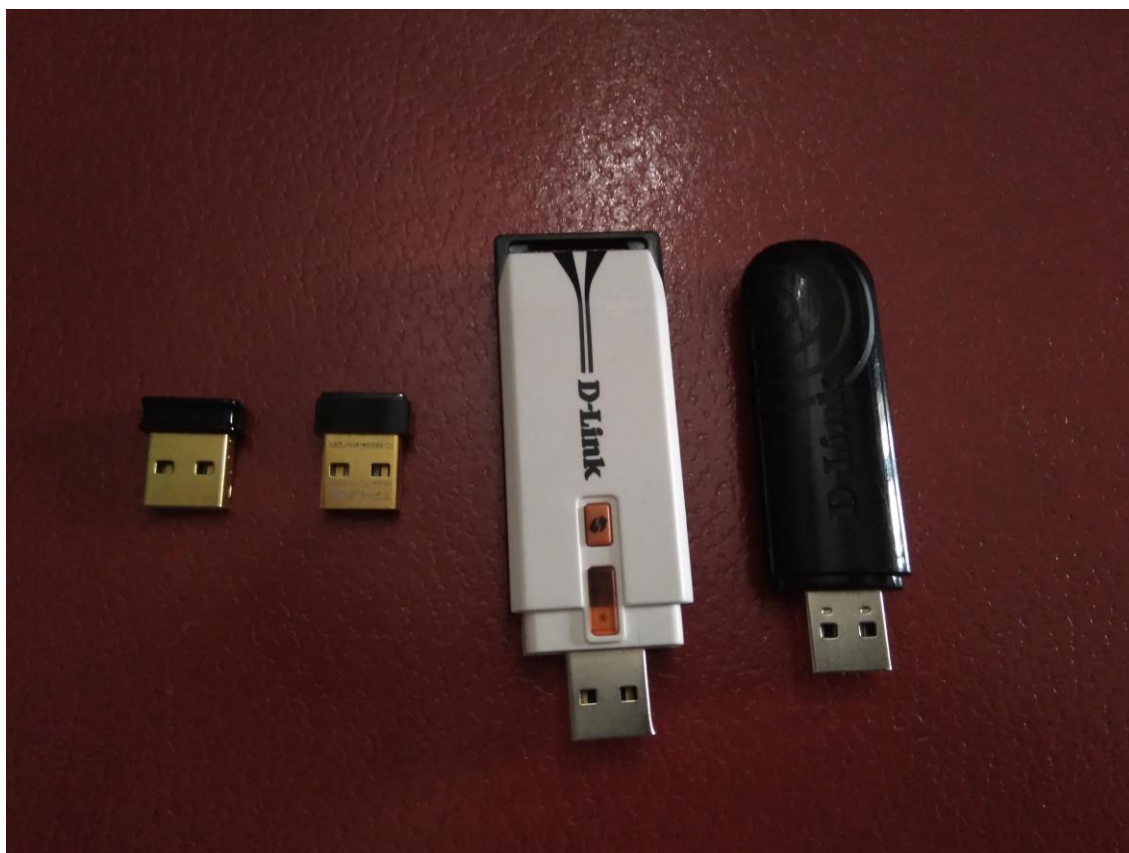


Figura 3-1. De izquierda a derecha EW7811Un, TL-WN725N, DWA-160 A2 y DWA-160 C1

A continuación se muestra una tabla comparativa de los dispositivos anteriormente mencionados en las que se comparan sus características más relevantes. Los datos han sido obtenidos de la propia web de los fabricantes (en la propia tabla se proporcionan enlaces a las webs de los mismos), así como de los datasheets de los dispositivos.

Tabla 3-1. Comparativa de los adaptadores de red

| Nombre    | Link                    | Certificaciones  | Chipset    | Driver<br>Windows           | Driver<br>Linux | Tasa<br>(802.11n) | Precio |
|-----------|-------------------------|------------------|------------|-----------------------------|-----------------|-------------------|--------|
| EW-7811Un | <a href="#">Edimax</a>  | 802.11b/g/n      | RTL8188CUS | Si                          | Si              | >150Mbps          | 7,65€  |
|           |                         | QoS WMM          |            |                             |                 | (40MHz)           |        |
|           |                         | WMM-PS           |            |                             |                 | >72Mbps           |        |
|           |                         | WEP/WPA/2<br>WPS |            |                             |                 | (20MHz)           |        |
| TL-WN725N | <a href="#">TP-Link</a> | 802.11b/g/n      | RTL8188EUS | Si                          | Si              | >150Mbps          | 11,99€ |
|           |                         | WEP/WPA/2        |            | [incluido<br>Windows<br>10] |                 | (40MHz)           |        |

|               |                        |  |                 |                                       |    |                    |        |
|---------------|------------------------|--|-----------------|---------------------------------------|----|--------------------|--------|
|               |                        | IEEE 802.1X  |                 | 10]                                   |    | >72Mbps<br>(20MHz) |        |
| DWA-160<br>A2 | <a href="#">D-Link</a> | 802.11a/b/g/n<br>802.11n (draft<br>2.0) Dualband<br>WEP/WPA/2/<br>WPS<br>QoS WMM<br>WMM-PS | AR9001U-<br>2NX | Si                                    | No | >300Mbps           | 35,07€ |
| DWA-160<br>C1 | <a href="#">D-Link</a> | 802.11a/b/g/n<br>802.11n (draft<br>2.0) Dualband<br>WEP/WPA/2/<br>WPS<br>QoS WMM<br>WMM-PS | RT5572          | Si<br><br>[incluido<br>Windows<br>10] | No | >300Mbps           | 34,49€ |

### 3.1.5 D-Link DAP 2590

Se trata de un punto de acceso inalámbrico orientado a entornos empresariales, y que dispone de una gran cantidad de características y funcionalidades más avanzadas que las que soportan los AP básicos.

Soporta 802.11 a/b/g/n, y funciona en las bandas de 2,4 y 5 GHz, así mismo puede ser alimentado mediante un cable de red (RJ-45), ya que también soporta 802.3af(PoE).

A nivel de seguridad soporta WEP, WPA, WPA2, así mismo incorpora un servidor RADIUS interno y permite el empleo de listas de control de acceso (ACL) basadas en direcciones MAC.

La gestión del punto de acceso es bastante sencilla y flexible, ya que incorpora varios mecanismos para la misma: SSH, telnet, un interfaz web y soporta SNMP v1,v2c y v3. En nuestros escenarios, se ha empleado el interfaz web por ser intuitivo y más cómodo para la configuración de las funcionalidades de QoS.

Dispone de 3 antenas, lo que permite utilizar hasta tres SSID diferentes en un mismo AP y soporta Wireless Distribution System (WDS), permitiendo así la interconexión inalámbrica de varios APs sin necesidad de un cable troncal que los conecte.

Proporciona tasas superiores a los 300Mbps en ambas bandas de frecuencia, soporta el control de tráfico y calidad de servicio con la certificación de Wi-Fi Multimedia, por ello es un punto de acceso ideal para aplicaciones de audio video y voz. También ofrece la posibilidad de realizar balanceo de carga, así como de definir reglas para la priorización del tráfico en función de sus puertos y el mapeo de DSCP.

Este producto utiliza programación de software realizada por terceros, incluyendo código software sujeto a la GNU General Public License ("GPL") o GNU Lesser General Public License ("LGPL").

Para más información a cerca del dispositivo puede consultarse la web del fabricante en la siguiente referencia.  
[15]



Figura 3-2. D-Link DAP-2590 [25]

### 3.1 Pruebas en sistemas operativos (Primera fase)

Para el desarrollo del proyecto inicialmente se planteó el empleo de Linux, pues el proyecto anterior se desarrolló en este sistema operativo. No obstante, cabe señalar que pjsip está disponible también en Windows, pero los scripts y utilidades que se desarrollaron e integraron en el proyecto anterior solo funcionan en Linux.

Sin embargo, tras varios problemas que a continuación se analizan en profundidad, se decidió el empleo de Windows para los escenarios de prueba.

#### 3.1.1 Linux

Debido a que los ordenadores del laboratorio de telemática disponen tanto de Debian como de CentOS 7, inicialmente se realizaron algunas pruebas en ambos sistemas operativos, pero finalmente se seleccionó Debian, concretamente la versión Wheezy 7.3.

La instalación de los drivers de Linux se realizó exclusivamente para los nanoadaptadores puesto que son los únicos para los que los fabricantes proporcionan drivers. Para instalarlos, en primer lugar descargamos los drivers de la página oficial de Edimax o TP-Link según corresponda.

##### 3.1.1.1 EW-7811Un

Una vez descargado, descomprimos el fichero. Para ello nos situamos en el directorio en el que se encuentre el fichero y ejecutamos:

```
root@debian:/home/dit/Descargas# unzip EW-7811Un_Linux_driver_v1.0.0.5.zip
```

Dentro de la carpeta descomprimida encontramos a su vez una carpeta llamada `RTL8188C_8192C_USBlinux_v4.0.2_9000.20130911`, nos situamos dentro de la misma y ejecutamos:

```
root@debian:/home/dit/Descargas/EW-7811Un_Linux_driver_v1.0.0.5/RTL8188C_8192C_USBlinux_v4.0.2_9000.20130911# ./install.sh
```

Cuando veamos una salida como esta por el terminal, la instalación habrá finalizado, y ya se puede conectar el adaptador inalámbrico:

```
#####
The Setup Script is completed !
#####
```

Dentro de la carpeta anteriormente mencionada encontramos también la documentación del driver (document) y el dispositivo así como las Wireless Tools for Linux [22] (wireless\_tools) y WpaSupplicant (wpa\_supplicant\_hostapd) por si no se dispone de las mismas en el equipo. Ambas incorporan una guía con los pasos a seguir que facilita el proceso de instalación.

### 3.1.1.2 TL-WN725N

Una vez descargado, descomprimos el fichero. Para ello nos situamos en el directorio en el que se encuentre el fichero y ejecutamos:

```
root@debian:/home/dit/Descargas# unzip TL-WN725N_V2_150911.zip
```

Dentro de la carpeta descomprimida encontramos a su vez una carpeta llamada Driver, nos situamos dentro de la misma y ejecutamos:

```
root@debian:/home/dit/Desktop/TL-WN725N_V2_150911/Driver# make
```

Cuando termine la ejecución del Makefile el driver está instalado y podemos conectar el adaptador.

En la carpeta que se crea al descomprimir el fichero encontramos un PDF (installation for Linux Driver\_RTL.pdf) que contiene toda la documentación del dispositivo y del driver.

Para la gestión de los interfaces inalámbricos en Linux se han empleado principalmente los comandos de Wireless Tools for Linux. Se trata de un conjunto de herramientas que permiten manipular las Wireless Extensions de Linux mediante un interfaz de línea de comandos. Wireless Extensions se define como una API genérica que permite a un driver exponer estadísticas y opciones de configuración de las redes de área local inalámbrica en el entorno de usuario.

A pesar de que existen otras herramientas para la gestión de las Wireless Extensions, las Wireless Tools son la implementación de referencia.

Las herramientas que incluye son las siguientes

- iwconfig permite manipular los parámetros básicos de los interfaces inalámbricos.
- iwlist permite realizar escaneos y listar frecuencias, tasas, tipos de encriptación, etc.
- iwspy permite obtener la calidad del enlace por nodo.
- iwpriv permite manipular las WirelessExtensions específicas de un driver (privadas).
- ifrename permite renombrar interfaces basándose en varios criterios estáticos.

Principalmente nos interesan iwconfig e iwpriv puesto que son los comandos que nos permiten actuar sobre los modos de ahorro de energía.

Iwconfig permite actuar sobre el modo de ahorro de energía tradicional., permite modificar el periodo entre “despertares” de la estación, establecer valores máximos y mínimos así como valores de timeout y activar o desactivar el modo de ahorro de energía.

```
iwconfig eth0 power period 2
iwconfig eth0 power 500m unicast
iwconfig eth0 power timeout 300u all
iwconfig eth0 power saving 3
iwconfig eth0 power off
iwconfig eth0 power min period 2 power max period 4
```

Figura 3-3. Ejemplos de comandos con iwconfig.

Puesto que no nos interesa el modo de ahorro de energía tradicional, nos centramos en iwpriv. Cuando se

ejecuta el comando sin argumentos obtenemos una lista de los ioctls privados de cada interfaz.

```

root@debian:/home/dit/Descargas/TL-WN725N_V2_150911# iwpriv
lo          no private ioctls.

wlan0      Available private ioctls :
write      (8BE0) : set 2047 char & get 0
read       (8BE1) : set 2047 char & get 16 char
driver_ext (8BE2) : set 0 & get 0
mp_ioctl   (8BE3) : set 0 & get 0
apinfo     (8BE4) : set 1 int & get 0
setpid     (8BE5) : set 2 int & get 0
wps_start  (8BE6) : set 1 int & get 0
get_sensitivity (8BE7) : set 1 int & get 0
wps_prob_req_ie (8BE8) : set 1 int & get 0
wps_assoc_req_ie (8BE9) : set 1 int & get 0

```

Figura 3-4. Algunos Ioctls privados del adaptador EW7811Un.

Por desgracia, los ioctls privados que ofrece para el adaptador EW-7811Un no permiten actuar sobre el modo de ahorro de energía de WMM, ni siquiera permite activarlo y el adaptador TL-WN725N no muestra ningún ioctl privado.

### 3.1.1.3 DWA-160

Debido a que los otros dos adaptadores no ofrecen drivers para Linux fue necesario buscar una forma de adaptar los drivers de Windows. Para ello se empleó ndiswrapper [16] un encapsulador o envoltura (wapper) libre de controladores de dispositivos, que permite el uso de la mayoría de las tarjetas de red inalámbricas en sistema operativo GNU utilizando la APIs del kernel de Windows, enlazando de forma dinámica el controlador de Windows a esta implementación.

Para instalar los drivers de los adaptadores DWA-160 (ambas revisiones) en primer lugar descomprimos el fichero descargado desde la web del fabricante:

```

root@debian:/home/dit/Descargas# unzip DWA-160_drvWIN_revC_3-04_eu_multi_20150916.zip

```

Dentro de la carpeta descomprimida encontramos a su vez una carpeta llamada Drivers, y en esta hay varias carpetas, cada una para una versión de Windows y para una arquitectura concreta (x64/x86):

```

root@debian:/home/dit/Descargas/DWA-160 C1 v3.04/Drivers# ls
Win10x64 Win7x64 Win8BLUEx64 Win8x64 x64 XP_VISTAX64
Win10x86 Win7x86 Win8BLUEx86 Win8x86 x86 XP_VISTAX86

```

Nos situamos en la carpeta del driver en cuestión y ejecutamos el siguiente comando, donde el fichero acabado en .inf es el driver:

```

root@debian:/home/dit/Descargas/DWA-160 C1 v3.04/Drivers/Win8x86# ndiswrapper -i netr28u.inf
installing netr28u ...

```

Una vez que el driver está correctamente instalado, cargamos el módulo al sistema:

```

root@debian:/home/dit/Descargas/DWA-160 C1 v3.04/Drivers/Win8x86# depmod -a
root@debian:/home/dit/Descargas/DWA-160 C1 v3.04/Drivers/Win8x86# modprobe ndiswrapper
root@debian:/home/dit/Descargas/DWA-160 C1 v3.04/Drivers/Win8x86#

```

Configuramos modprobe para que se cargue ndiswrapper cuando la interfaz inalámbrica esté activada:

```

root@debian:/home/dit/Descargas/DWA-160 C1 v3.04/Drivers/Win8x86# ndiswrapper -m
adding "alias wlan0 ndiswrapper" to /etc/modules.conf ...

```

Por último, configuramos el sistema para que al arrancar este cargue el módulo ndiswrapper, para ello abrimos y editamos el archivo /etc/modules y añadimos ndiswrapper al final del archivo y guardamos. Podemos comprobar cómo el módulo ndiswrapper que ejecuta el controlador del adaptador inalámbrico esta correctamente cargado podemos ejecutar el comando lsmod.

Una vez realizada la instalación, los dispositivos ya funcionan pero no muestran ioctls privados por lo que tampoco podremos activar ni modificar el funcionamiento del modo de ahorro de energía.



Debido a la imposibilidad de activar el funcionamiento del modo de ahorro de energía WMM-PS de los adaptadores inalámbricos en Linux se decidió abandonar la investigación del funcionamiento de los drivers y probar en Windows.

### 3.1.2 Windows

#### 3.1.2.1 Windows 7

La instalación de los drivers en Windows es bastante más sencilla, pues se limita a abrir el .zip descargado y ejecutar el asistente de instalación (setup.exe) que nos guiará para realizar la instalación del dispositivo en todos los casos.

Una vez instalados los drivers del dispositivo, es necesario activar el soporte para QoS de los mismos, para ello accedemos a la configuración del adaptador en:

Panel de control>Redes e Internet>Conexiones de red>Conexión de red inalámbrica X>Propiedades>Configurar>Opciones Avanzadas

Dentro de esta pestaña podremos encontrar varias opciones avanzadas, que también varían en función del adaptador, pero la principal que nos interesa es QoS Support, cuyo valor debe ponerse a SupportQoS en el caso de los nanoadaptadores.

Para el caso de los adaptadores DWA-160 las opciones avanzadas varían, y en concreto para el rev.C1 la opción a activar es Ambiente de Juego/Multimedia, mientras que el rev.A2 no proporciona ninguna opción para activar la QoS.

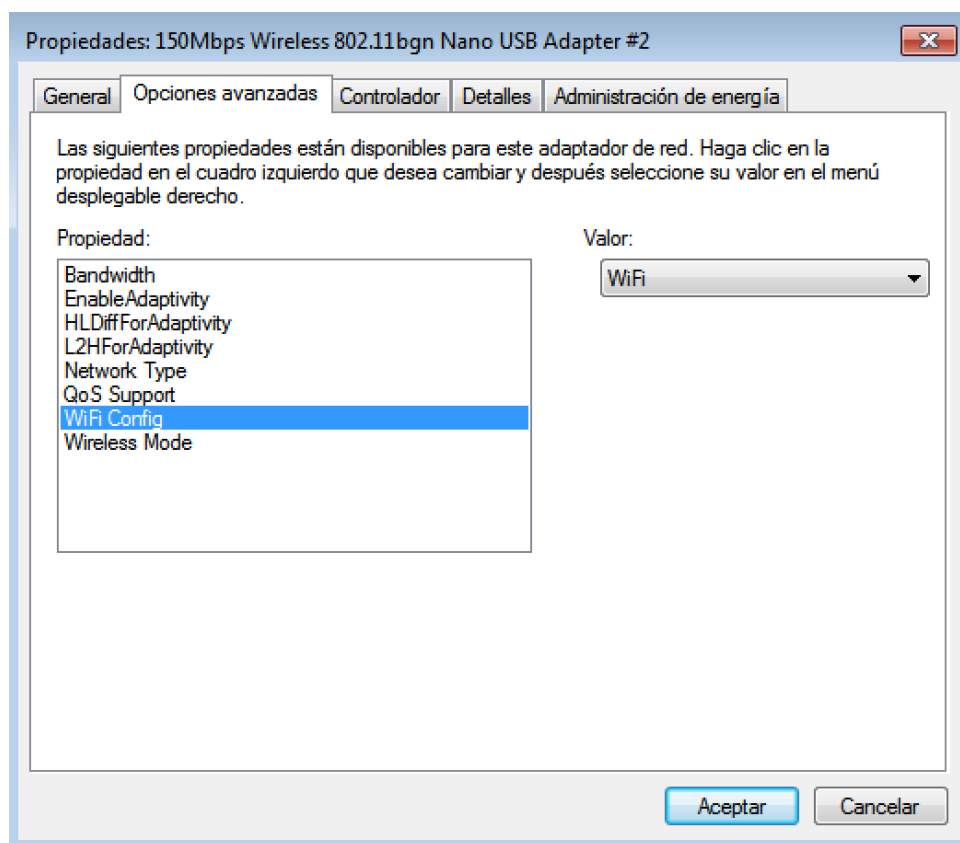


Figura 3-5. Configuración avanzada del adaptador inalámbrico

Posteriormente, y para poder definir políticas de QoS en el equipo es necesario modificar un parámetro del registro. Para ello se emplea el editor de registro de Windows (regedit), y se accede a la siguiente ruta:

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\Tcpip\QoS*

Comprobamos que el campo “Do not use NLA” tiene el valor 1. En el caso de que no existiera la entrada en el registro, será necesario crearla. Se proporciona un enlace con información a cerca de la manipulación del registro de Windows por si fuera necesario. [24]

La modificación de este parámetro es esencial pues aunque creemos políticas que asignen valores de QoS a los paquetes salientes del equipo, si no se iguala 1 no se le asigna el valor correspondiente al DSCP y en consecuencia no se realiza el mapeo correspondiente por la capa MAC 802.11e.

Para crear las políticas de QoS en primer lugar accedemos Consola de Gestión de Microsoft (Inicio>Ejecutar>MMC) y una vez que se ha abierto pulsamos control+m para agregar un nuevo complemento. A continuación seleccionamos Editor de objetos de directiva de grupo del desplegable de la izquierda, y pulsamos el botón Agregar y posteriormente el botón Aceptar.

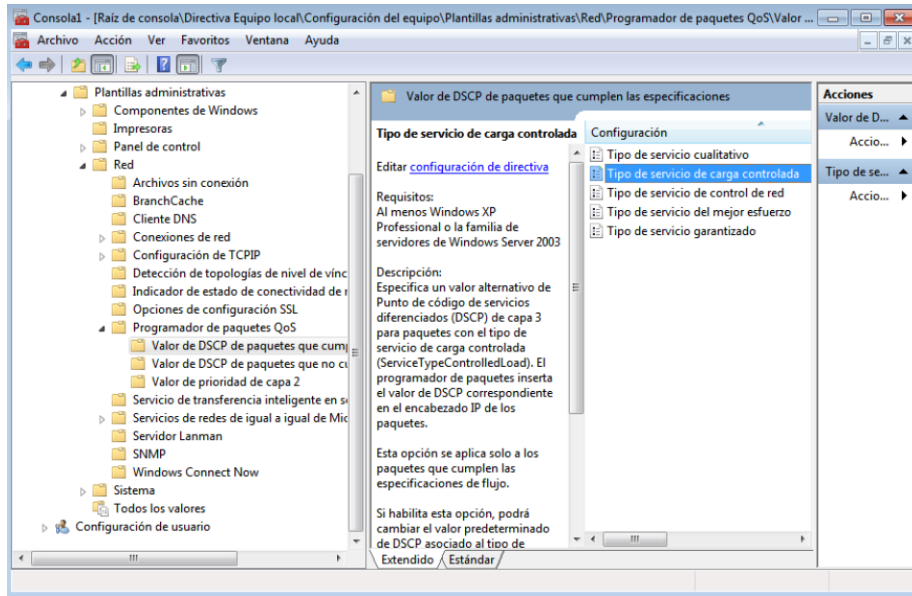


Figura 3-6. Consola de gestión de Microsoft

Una vez hecho esto, la nueva directiva aparecerá a la izquierda de la consola de gestión. Si pulsamos encima de ella podremos ver que aparecen varios desplegables. Debemos pulsar los desplegables siguiendo la siguiente ruta:

Configuración del equipo>Plantillas Administrativas>Red>Programador de paquetes de QoS

Una vez que hayamos hecho esto encontramos tres carpetas: Valor de DSCP de paquetes que cumplen las especificaciones, Valor de DSCP de paquetes que no cumplen las especificaciones, y valor de prioridad de capa 2.

Dentro de la primera de ella encontramos varias configuraciones, cada una de ella para un tipo de servicio: cualitativo, carga controlada, control de red, mejor esfuerzo y garantizado. Haciendo doble click encima de ellas podemos editarlas, concretamente habilitarlas, marcando Habilitada y dejando el valor por defecto del DSCP.

Para el valor de CoS (Class of Service) de nivel dos, seguimos un procedimiento similar pero en la carpeta valor de prioridad de capa 2.

Establecemos el funcionamiento de la priorización en ambas capas porque algunos dispositivos emplean el DSCP para el mapeo, mientras que otros utilizan directamente la CoS de nivel 2.

| Layer 3 / Layer 2 / WMM Mapping |              |                  |         |                     |
|---------------------------------|--------------|------------------|---------|---------------------|
| DSCP Field Hex/Bin/Dec          | Layer 2 Prio | Traffic Type     | Acronym | WMM Access Category |
| 0x38 / 111000 / 56              | 7            | Network Control  | NC      | AC_VO               |
| 0x30 / 110000 / 48              | 6            | Voice            | VO      | AC_VO               |
| 0x28 / 101000 / 40              | 5            | Video            | VI      | AC_VI               |
| 0x20 / 100000 / 32              | 4            | Controlled Load  | CL      | AC_VI               |
| 0x18 / 011000 / 24              | 3            | Excellent Effort | EE      | AC_BE               |
| 0x10 / 010000 / 16              | 2            | Spare            | --      | AC_BK               |
| 0x08 / 001000 / 8               | 1            | Background       | BK      | AC_BK               |
| 0x00 / 000000 / 0               | 0            | Best Effort      | BE      | AC_BE               |

Figura 3-7. Mapeo de DSCP/COS/WMM AC [17]

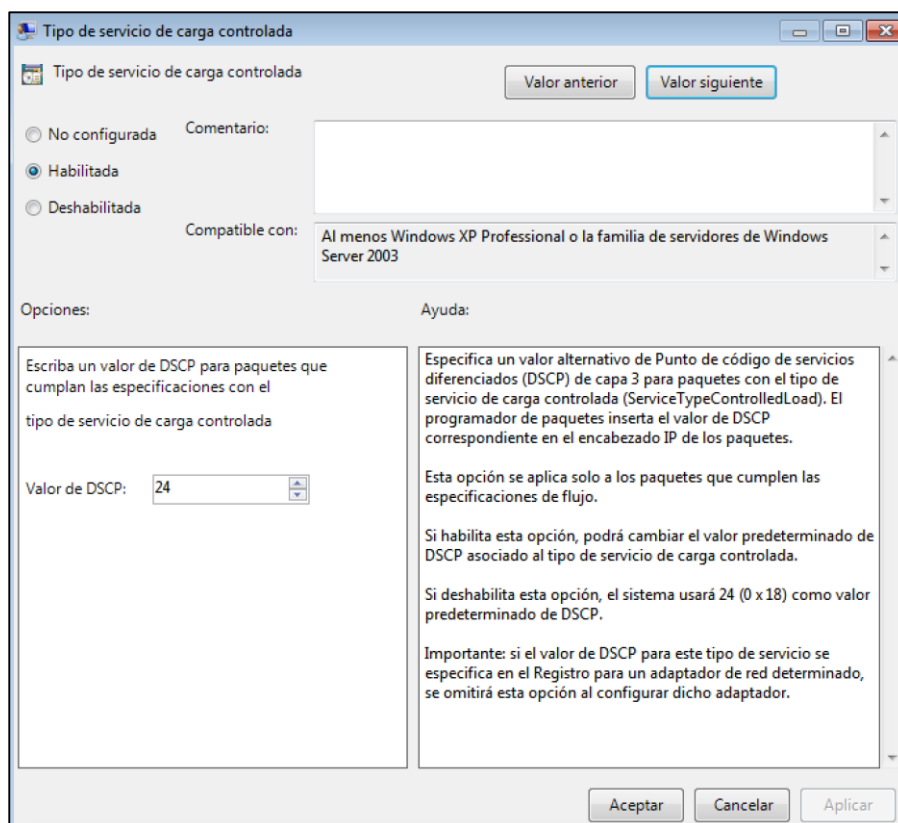


Figura 3-8. Editando la configuración de un tipo de servicio

Si quisiéramos asignar un valor concreto de DSCP a una determinada aplicación, también puede realizarse desde la consola de gestión, y de esta forma, todos los paquetes generados por un determinado ejecutable o dirigidos a una URL son marcados con el mismo valor de DSCP.

Para ello accedemos a la siguiente ruta en los desplegados de la consola:

Configuración del equipo>Configuración de Windows>QoS basada en directiva

Haciendo click derecho en QoS basada en directiva podemos nos permite añadir una nueva directiva de QoS especificando el valor de DSCP, el nombre del ejecutable de la aplicación o bien la URL que atienden en caso de tratarse de un servidor HTTP las IPs de origen y destino, el protocolo de transporte y los puertos empleados.

Para la realización de pruebas se creó una directiva de QoS para rtpptools, herramienta que posteriormente analizaremos en profundidad, asignándole el valor de DSCP 56, correspondiente a AC\_VO y a un valor de CoS 7.

No obstante, es necesario señalar que en algunas versiones de Windows 7, es necesario modificar algunos parámetros más en el registro para poder utilizar las directivas de QoS definidas por el usuario. Los valores a modificar se encuentran ya incluidos en el fichero de registro */Modificaciones/Win7DSCPOverride.REG* del CD adjunto, basta con ejecutarlo para que los cambios se hagan efectivos.

Por último es necesario modificar las preferencias de energía del sistema accediendo a Panel de control>Opciones de energía>Cambiar la configuración del plan>Cambiar la configuración avanzada de energía. Una vez que se abre el menú de configuración avanzada es necesario desplegar la opción de *Configuración de adaptador inalámbrico* y a continuación *Modo de ahorro de energía* para seleccionar el modo *Ahorro de energía máximo*.

Es muy importante el seguimiento de estos pasos para garantizar el funcionamiento de WMM-PS en aquellos dispositivos que lo soporten, ante cualquier duda se puede consultar el Anexo A en el que se explica detalladamente la configuración de los dispositivos.

#### 3.1.2.1.1 Prueba 1

Tras toda esta configuración se realizaron múltiples capturas de paquetes con el objetivo de comprobar si la función de QoS estaba activa, y si las categorías se estaban asignando adecuadamente.

Para realizar estas capturas se empleó un escenario de pruebas sencillo, con el AP de D-Link conectado a un Switch por Ethernet, y se configuró el AP para que soportase QoS siguiendo los pasos especificados en el Anexo B.

Como fuente y Sumidero de tráfico se emplearon dos ordenadores de sobremesa, el PC 1, en el que se utilizaron los adaptadores USB DWA-160 E rev.C1 y rev.A2, y que ejecutaba rtpsend, la aplicación encargada de generar el tráfico simulado de VoIP, y el PC2, conectado por Ethernet al switch, que también ejecutaba rtpsend. Ambos equipos ejecutaban también rtpdump, la aplicación encargada de recibir el tráfico generado por rtpsend.

Se empleó también un Macbook Pro retina 13, que utilizó su tarjeta de red interna (Airport Extreme) para capturar paquetes en modo monitor.

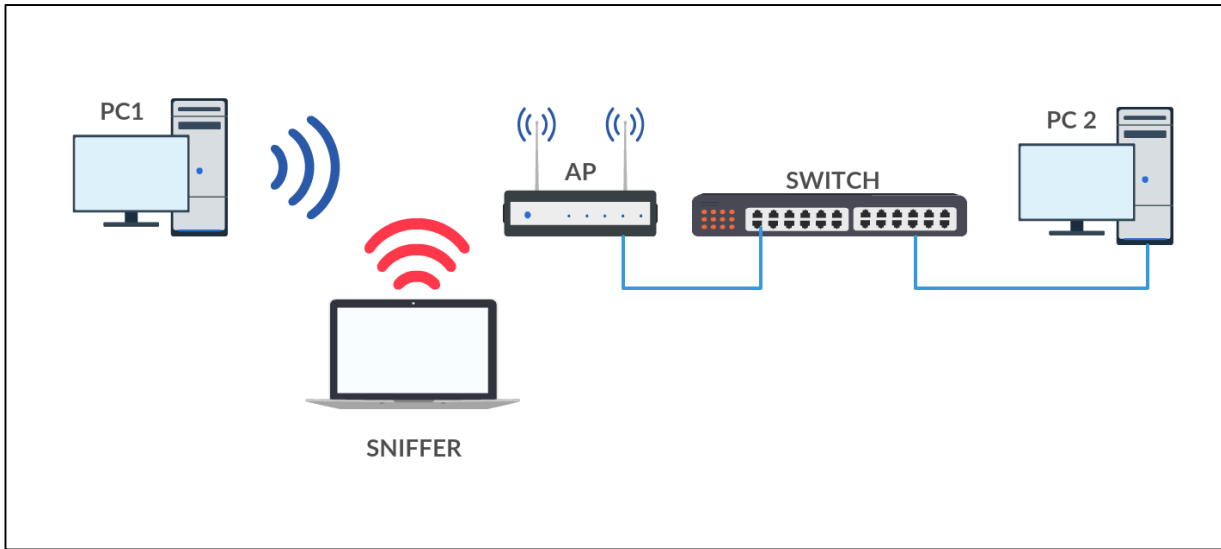


Figura 3-9. Topología adoptada en la prueba

La captura que se muestra a continuación corresponde con una trama Beacon emitida por el AP de D-Link en la que además de los parámetros de QoS para las diferentes categorías de acceso puede observarse como dentro del campos de información de QoS WMM el modo de ahorro de energía U-APSD se encuentra activado.

```

▼ Tag: Vendor Specific: Microsof: WMM/WME: Parameter Element
  Tag Number: Vendor Specific (221)
  Tag length: 24
  OUI: 00-50-f2
  Vendor Specific OUI Type: 2
  Type: WMM/WME (0x02)
  WME Subtype: Parameter Element (1)
  WME Version: 1
  ▼ WME QoS Info: 0x82
    1... .... = U-APSD: Enabled
    .... 0010 = Parameter Set Count: 0x02
    .000 .... = Reserved: 0x00
  Reserved: 00
  ▶ Ac Parameters ACI 0 (Best Effort), ACM no, AIFSN 3, ECWmin 4, ECWmax 10, TXOP 0
  ▶ Ac Parameters ACI 1 (Background), ACM no, AIFSN 7, ECWmin 4, ECWmax 10, TXOP 0
  ▶ Ac Parameters ACI 2 (Video), ACM no, AIFSN 2, ECWmin 3, ECWmax 4, TXOP 94
  ▶ Ac Parameters ACI 3 (Voice), ACM no, AIFSN 2, ECWmin 2, ECWmax 3, TXOP 47

```

Figura 3-10. Detalle de un Beacon emitido por el AP

En la siguiente captura podemos observar que el mapeo de DSCP a la AC de voz se está realizando correctamente en el paquete que transmite el adaptador de D-Link al AP del mismo fabricante.

```
IEEE 802.11 QoS Data, Flags: .p....F.
  Type/Subtype: QoS Data (0x0028)
  ▶ Frame Control Field: 0x8842
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: Apple_86:96:61 (20:c9:d0:86:96:61)
    Destination address: Apple_86:96:61 (20:c9:d0:86:96:61)
    Transmitter address: HuaweiTe_ee:bf:42 (00:25:68:ee:bf:42)
    Source address: D-LinkIn_b4:88:74 (70:62:b8:b4:88:74)
    BSS Id: HuaweiTe_ee:bf:42 (00:25:68:ee:bf:42)
    STA address: Apple_86:96:61 (20:c9:d0:86:96:61)
    .... .... 0000 = Fragment number: 0
    1000 1011 1011 .... = Sequence number: 2235
  ▼ Qos Control: 0x0007
    .... .... 0111 = TID: 7
    [.... .... .111 = Priority: Network Control (Voice) (7)]
    .... .... 0 .... = EOSP: Service period
    .... .... .00. .... = Ack Policy: Normal Ack (0x0000)
    .... .... 0... .... = Payload Type: MSDU
  ▼ 0000 0000 .... .... = QAP PS Buffer State: 0x0000
    .... ..0. .... .... = Buffer State Indicated: No
  ▶ CCMP parameters
```

Figura 3-11. Mapeo correcto DSCP/AC

Sin embargo, en la captura del Association Request enviado por la estación al AP se pudo observar que no se solicita el empleo de WMM-PS para la categoría de acceso de voz, de hecho, no se solicita para ninguna categoría.

```
▼ Tag: Vendor Specific: Microsof: WMM/WME: Information Element
  Tag Number: Vendor Specific (221)
  Tag length: 7
  OUI: 00-50-f2
  Vendor Specific OUI Type: 2
  Type: WMM/WME (0x02)
  WME Subtype: Information Element (0)
  WME Version: 1
  ▼ WME QoS Info: 0x00
    .00. .... = Max SP Length: WMM AP may deliver all buffered frames (MSDUs and MMPDUs) (0x00)
    .... 0... = AC_BE: non-WMM PS
    .... .0.. = AC_BK: non-WMM PS
    .... ..0. = AC_VI: non-WMM PS
    .... ...0 = AC_VO: non-WMM PS
    0..0 .... = Reserved: 0x00
```

Figura 3-12. Detalle de un Association Request

Las pruebas realizadas con las dos revisiones del modelo DWA-160 arrojaron resultados similares y permitieron llegar a la conclusión de que el modo de ahorro de energía que se buscaba no estaba soportado por los dispositivos.

### 3.1.2.1.2 Prueba 2

Debido a la imposibilidad de activar WMM-PS con el adaptador DWA-160 E rev.C1 y rev.A2, se decidió probar con los nanoadaptadores, utilizando un escenario igual al de la prueba anterior. No obstante, es necesario indicar que fue necesario investigar en profundidad el registro de ambos adaptadores con Regedit para poder activarlo. La ruta exacta es la siguiente, donde XX puede variar en función del equipo e indica un interfaz diferente.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\sControl\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\00XX\StaUapsd
```

Basta con poner el valor a 1 para ambos adaptadores (por defecto se encuentra a 0).

Es preciso comentar que aunque muchos de los parámetros que aparecen en el registro pueden ser modificados desde la ventana de configuración de interfaz de red, este parámetro en concreto no aparece en la misma, por ello se realizó la modificación manualmente.

Tras modificarlo se puede observar como en la asociación el adaptador indica que al menos la categoría de

acceso está activa para WMM-PS

```

▼ WME QoS Info: 0x01
  .00. .... = Max SP Length: WMM AP may deliver all buffered frames (MSDUs and MMPDUs) (0x00)
  .... 0... = AC_BE: non-WMM PS
  .... .0.. = AC_BK: non-WMM PS
  .... ..0. = AC_VI: non-WMM PS
  .... ...1 = AC_VO: WMM delivery and trigger enabled
  0..0 .... = Reserved: 0x00
  
```

Figura 3-13. Detalle de un Association Request

Sin embargo, durante el intercambio de tramas, en las que son enviadas por los nanoadaptadores, nunca encontramos el flag de PWR MGT a 1 solicitando la información almacenada en el AP.

```

▼ Flags: 0x01
  .... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
  .... .0.. = More Fragments: This is the last fragment
  .... 0... = Retry: Frame is not being retransmitted
  ...0 .... = PWR MGT: STA will stay up
  ..0. .... = More Data: No data buffered
  .0.. .... = Protected flag: Data is not protected
  0... .... = Order flag: Not strictly ordered
  
```

Figura 3-14. Flags de una cabecera 802.11

Además, en las tramas que el AP envía al nanoadaptador tampoco encontramos el bit EOSP a 1 indicando el final del periodo de servicio asociado al adaptador nunca.

```

..... ..0111 = TID: 7
[..... ..0111 = Priority: Network Control (Voice) (7)]
..... ..0 .... = EOSP: Service period
..... ..00. .... = Ack Policy: Normal Ack (0x0000)
..... ..0... .... = Payload Type: MSDU
0000 0000 ..... = QAP PS Buffer State: 0x0000
  
```

Figura 3-15. Detalle del Campo QoS Control de una cabecera 802.11

Para ambos nanoadaptadores se obtuvieron resultados muy similares, concluyendo por tanto que el funcionamiento de WMM-PS para los mismos no era el esperado, y que a pesar de indicar al AP que almacenase las tramas de voz en el buffer, los adaptadores no enviaban tramas trigger para desencadenar el envío de los datos almacenados.

## 3.2 Pruebas en sistemas operativos (Segunda Fase)

Tras la realización de las primeras pruebas se pudo constatar que ninguno de los adaptadores funcionaba con WMM-PS/U-APSD a pesar de ser indicado por el fabricante en sus especificaciones, por lo que se realizó una investigación con el objetivo de determinar los adaptadores a adquirir para poder proseguir con el proyecto.

Además se optó por utilizar únicamente Windows, ya que las soluciones ofrecidas para el mercado DSCP y la compatibilidad con las tarjetas facilitaban el desarrollo de las pruebas.

En la siguiente tabla se muestra una comparativa de los adaptadores de red que en un principio garantizarían el correcto funcionamiento de WMM-PS y que se adquirieron para el desarrollo del proyecto.

Tabla 3-2. Comparativa de los adaptadores de red

| Nombre | Link | Certificaciones | Chipset | Driver<br>Windows | Driver<br>Linux | Tasa | Precio |
|--------|------|-----------------|---------|-------------------|-----------------|------|--------|
|--------|------|-----------------|---------|-------------------|-----------------|------|--------|



|                    |                        |   |                 |                                   |    |                                 |        |
|--------------------|------------------------|---|-----------------|-----------------------------------|----|---------------------------------|--------|
|                    |                        | 802.11a/b/g/n/ac  |                 |                                   |    |                                 |        |
| PCE-AC56<br>(PCIe) | <a href="#">Asus</a>   | QoS WMM<br>WMM-PS<br>WEP/WPA/2<br>WPS                   | BCM4352         | Si<br>[Incluido<br>Windows<br>10] | Si | Hasta<br>1300Mbps<br>(20/40MHz) | 49,95€ |
|                    |                        | 802.11a/b/g/n/ac  |                 |                                   |    |                                 |        |
| PCE-AC68<br>(PCIe) | <a href="#">Asus</a>   | QoS WMM<br>WMM-PS<br>WEP/WPA/2<br>WPS                   | BCM4360         | Si<br>[Incluido<br>Windows<br>10] | Si | Hasta<br>1300Mbps<br>(20/40MHz) | 75,00€ |
|                    |                        | 802.11a/b/g/n/ac  |                 |                                   |    |                                 |        |
| AC-7260<br>(PCIe)  | <a href="#">Intel</a>  | Bluetooth 4.0<br>QoS WMM<br>WEP/WPA/2/<br>WPS<br>WMM-PS | Propio de Intel | Si<br>[Hasta<br>Windows<br>8]     | Si | 300/867Mbps<br>(20/40MHz)       | 29,98€ |
|                    |                        | 802.11a/b/g/n   |                 |                                   |    |                                 |        |
| EW-7718Un          | <a href="#">Edimax</a> | WEP/WPA/2/<br>WPS<br>QoS WMM<br>WMM-PS                  | RT2870          | Si<br>[incluido<br>Windows<br>10] | No | >300Mbps                        | 19,60€ |

### 3.2.1 Windows 7

#### 3.2.1.1 Asus PCE-AC56



Figura 3-16. Asus PCE-AC56 con soporte de antenas

Se trata de un adaptador de red con conector PCI Express de altas prestaciones, que cumple con los estándares 802.11a/b/g/n y el nuevo estándar 802.11ac e incorpora dos antenas que pueden montarse directamente sobre la tarjeta o en una peana magnética para su montaje sobre la carcasa de los PCs de sobremesa.

Incorpora un disipador de dimensiones considerables para extraer el calor y asegura un funcionamiento continuo y estable del adaptador. Cuenta también con la tecnología AirRadar de Asus, que le permite localizar la posición de los puntos de accesos gracias a sus antenas.

Cuenta con la tecnología Turbo QAM, que permite incrementar hasta los 200Mbps por flujo en la banda de 2,4 GHz con canales de 40MHz utilizando 256-QAM en lugar de la 64-QAM empleada por el estándar 802.11n.

El disco de instalación, además de los drivers, contiene una aplicación para la gestión del adaptador y permite modificar algunas características (normales y avanzadas) de la misma como por ejemplo la habilitación de Turbo QAM. No obstante, dentro de las características que podemos modificar no se encuentra el funcionamiento de WMM-PS.





Figura 3-17. Detalle del asistente de configuración de Asus

Por ello se investigaron las opciones de configuración que ofrecía la tarjeta a través del interfaz de configuración del adaptador de Windows.

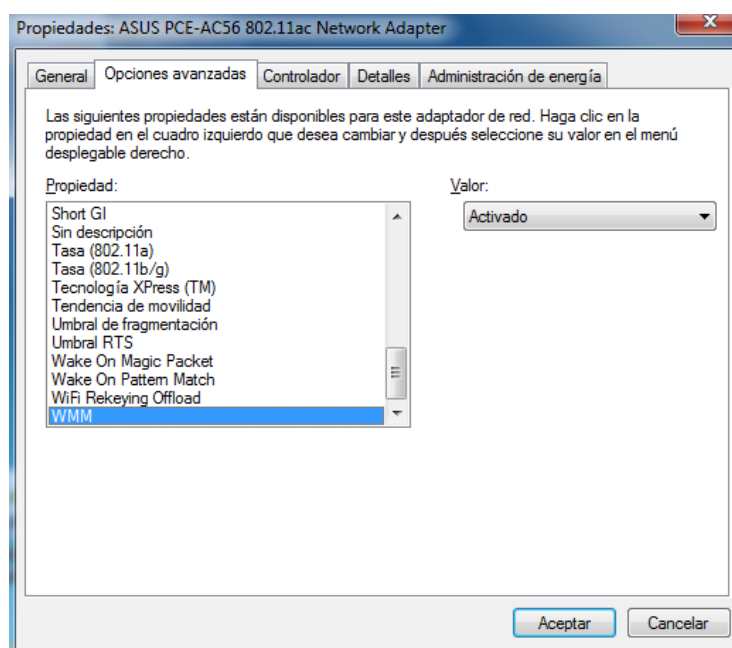


Figura 3-18. Detalle del interfaz de configuración de Windows

Como se puede observar en la figura 3-17, es posible activar WMM, y otras opciones entre las que cabe destacar *Ahorro Máximo de Energía*, que debe activarse para el funcionamiento de WMM-PS.

Una vez configurada, se repitieron las pruebas realizadas en la primera fase, con la misma configuración empleada, resultando insatisfactorias ya que en la asociación el adaptador no ponía los flags de las AC a 1 para realizar WMM-PS.

|                      |   |
|----------------------|---|
| ▼ WME QoS Info: 0x00 |   |
| .00. ....            | = Max SP Length: WMM AP may deliver all buffered frames (MSDUs and MMPDUs) (0x00) |
| .... 0...            | = AC_BE: non-WMM PS   |
| .... .0..            | = AC_BK: non-WMM PS   |
| .... ..0.            | = AC_VI: non-WMM PS   |
| .... ...0            | = AC_VO: non-WMM PS   |
| 0..0 ....            | = Reserved: 0x00  |

Figura 3-19. Detalle de un Association Request

Al igual que en los casos anteriores se investigaron los parámetros del registro que podían ser modificados, encontrando un único parámetro relacionado con el soporte de U-APSD (PowerSaveMode). Se realizaron varias pruebas variando el parámetro pero no se consiguió que funcionase.

|    |               |        |      |
|----|---------------|--------|------|
| ab | PowerSaveMode | REG_SZ | 0    |
| ab | ProviderName  | REG_SZ | ASUS |
| ab | PwrOut        | REG_SZ | 100  |
| ab | Rate          | REG_SZ | 0    |
| ab | RateA         | REG_SZ | 0    |

Figura 3-20. Detalle los registros del adaptador PC-AC56

### 3.2.1.2 Asus PCE-AC68



Figura 3-21. Asus PCE-AC68 con soporte de antenas.

Se trata de un adaptador de red con conector PCI Express de muy altas prestaciones, muy similar al PCE-AC56, pero que utiliza tres antenas y algunas mejoras.

Al igual que el adaptador previamente comentado incorpora AirRadar, Turbo QAM, incluye el software de Asus para su configuración y cumple los estándares 802.11ab/g/n/ac.

La principal diferencia es, como hemos comentado, el empleo de 3 antenas, que permite potenciar el funcionamiento de AirRadar, incrementar la cobertura y lo convierte en un adaptador de red inalámbrica idóneo para Streaming en HD y Gaming en línea.

En cuanto a las características de configuración, son exactamente iguales a las del PCE-AC56, incluyendo así mismo los registros, por lo que el resultado de las pruebas realizadas fue el mismo; no se consiguió activar el

modo WMM-PS. Por ello, también se descartó el PCE-AC68.

### 3.2.1.3 Intel AC-7260



Figura 3-22. Intel AC-7260 (Housing de fenvi)

Este adaptador de red incorpora además de las antenas para Wi-Fi, una antena para Bluetooth. Cumple los estándares 802.11ab/g/n/ac al igual que los anteriores, y Bluetooth 4.0 (incluye Bluetooth Low Energy).

Incorpora Intel Power Optimizer para la reducción del consumo energético, y es compatible con varias herramientas de gestión de Intel como el software PROSet/Wireless.

El adaptador que puede observarse en la figura 3-21 es concretamente un Fenvi FV-8801, pero el chipset, el controlador y el software que se proporcionan son básicamente los del AC-7260.

Para la configuración del adaptador se utilizó directamente el interfaz de configuración de Windows, ya que ofrecía explícitamente la posibilidad de activar el soporte de U-APSD y QoS, como se puede observar a continuación en la figura 3-22. Una vez activados, se procedió a la fase de pruebas que se recoge a continuación.

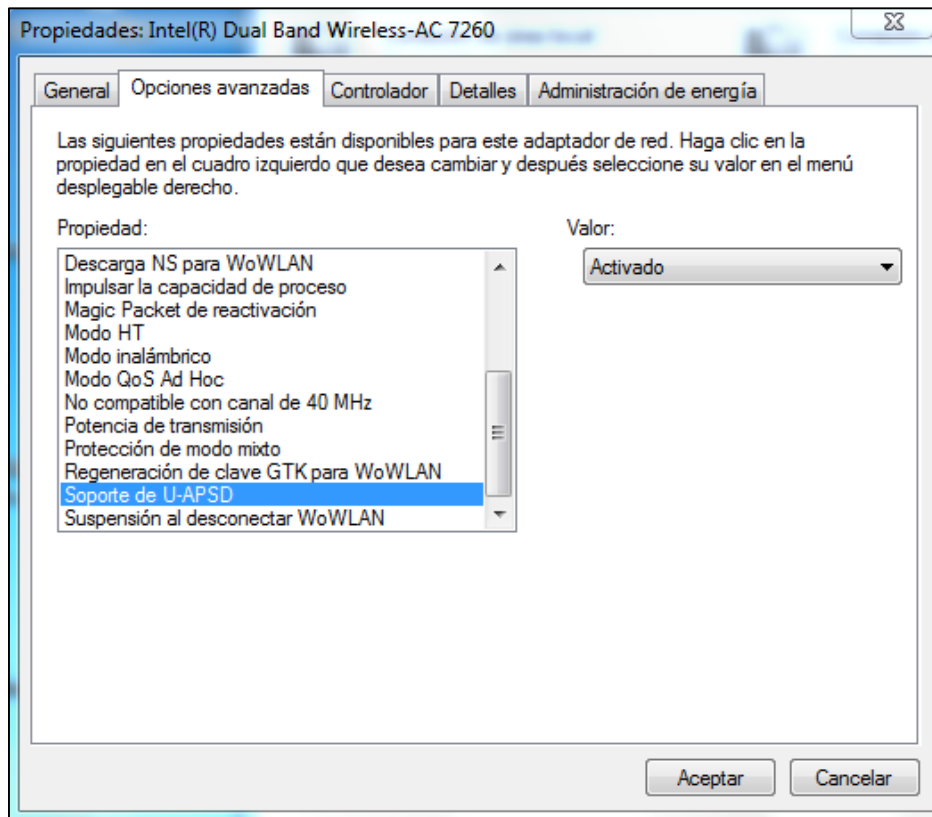


Figura 3-23. Propiedades del AC-7260 en el interfaz de configuración de Windows

En primer lugar, tras activar U-APSD, se puede observar como en el Association Request se levantan todos flags de todas las categorías de acceso.

```

WME QoS Info: 0x2f
.01. .... = Max SP Length: WMM AP may deliver a maximum of 2 buffered frames (MSDUs and MMPDUs) per USP (0x01)
.... 1... = AC_BE: WMM delivery and trigger enabled
.... .1.. = AC_BK: WMM delivery and trigger enabled
.... ..1. = AC_VI: WMM delivery and trigger enabled
.... ...1 = AC_VO: WMM delivery and trigger enabled
0..0 .... = Reserved: 0x00

```

Figura 3-24. Detalle de un Association Request

La secuencia concreta de mensajes que analizaremos para comprobar el funcionamiento de WMM-PS será la siguiente, donde la IP 192.168.0.14 corresponde a la estación con el interfaz inalámbrico y la IP 192.168.0.16 al equipo que se conecta al AP de forma cableada a través del switch. Nótese que hay tres tramas descendentes desde al AP a la estación por ser una de ellas una retransmisión.

| 192.168.0.14                                  | 192.168.0.16                                  | UDP    |
|---|---|--------|
| D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
|   | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
| 192.168.0.16                                  | 192.168.0.14                                  | UDP    |
| D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
|   | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
| 192.168.0.16                                  | 192.168.0.14                                  | UDP    |
|   | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
| D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
|   | D-LinkIn_0a:5d:88 (... IntelCor_c8:c6:07 (... | 802.11 |
| 192.168.0.16                                  | 192.168.0.14                                  | UDP    |

Figura 3-25. Secuencia de mensajes analizados en la captura

En primer lugar, en el mensaje ascendente del equipo 192.168.0.14, solicita el envío de las tramas almacenadas en la cola de la categoría AC\_VO con el bit PWR MGT a 1.

```

...1 .... = PWR MGT: STA will go to sleep
..0. .... = More Data: No data buffered
.0.. .... = Protected flag: Data is not protected
0... .... = Order flag: Not strictly ordered

```

Figura 3-26. Detalle de los flags de una cabecera 802.11

En el siguiente mensaje, en este caso enviado por el AP a la estación, se responde indicando que hay más información almacenada (More Data = 1) y que no ha acabado aún el periodo de servicio (EOSP = 0). Puede observarse cómo en este caso también se realiza correctamente el mapeado de DSCP a las AC.

```

.... ..10 = DS status: Frame from DS to a STA via AP(To DS: 0 From DS: 1) (0x02)
.... .0.. = More Fragments: This is the last fragment
.... 0... = Retry: Frame is not being retransmitted
...0 .... = PWR MGT: STA will stay up
..1. .... = More Data: Data is buffered for STA at AP
.0.. .... = Protected flag: Data is not protected
0... .... = Order flag: Not strictly ordered

```

Figura 3-27. Detalle de los flags de una cabecera 802.11

```

.... .... 0111 = TID: 7
[.... .... .111 = Priority: Network Control (Voice) (7)]
.... .... 0 .... = EOSP: Service period
.... .... .00. .... = Ack Policy: Normal Ack (0x0000)
.... .... 0... .... = Payload Type: MSDU
0000 0000 .... .... = QAP PS Buffer State: 0x0000

```

Figura 3-28. Detalle del campo QoS Control de una cabecera 802.11

Por último, en el siguiente mensaje, enviado nuevamente por el AP hacia la estación se indica que no hay más tramas en el buffer de la AC de voz (More Data = 0) y se indica que el periodo de servicio ha terminado (EOSP = 1).

```

.... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
.... .0.. = More Fragments: This is the last fragment
.... 0... = Retry: Frame is not being retransmitted
...0 .... = PWR MGT: STA will stay up
..0. .... = More Data: No data buffered
.0.. .... = Protected flag: Data is not protected
0... .... = Order flag: Not strictly ordered
    
```

Figura 3-29. Detalle de los flags de una cabecera 802.11

```

[.... ..111 = Priority: Network Control (Voice) (7)]
.... ..1 .... = EOSP: End of service period
.... ..00. .... = Ack Policy: Normal Ack (0x0000)
.... ..0... .... = Payload Type: MSDU
0000 0000 .... .... = QAP PS Buffer State: 0x0000
    
```

Figura 3-30. Detalle del campo QoS Control de una cabecera 802.11

Resumiendo, y para mostrarlo de una forma más esquemática, el intercambio de mensajes entre los dispositivos ha seguido la siguiente secuencia.

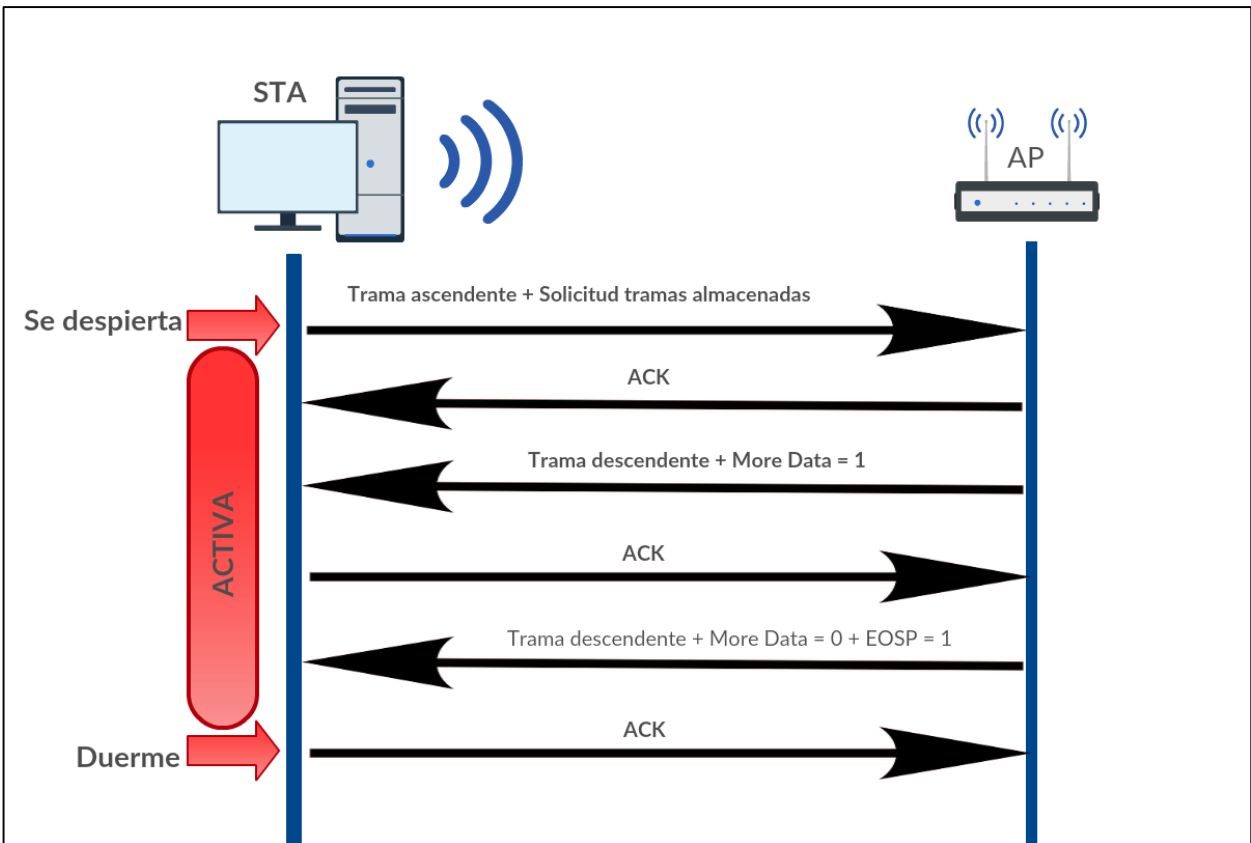


Figura 3-31. Esquema del funcionamiento de WMM-PS observado

A la vista del funcionamiento observado, puede concluirse que WMM-PS funciona correctamente para este adaptador inalámbrico, y por ello se eligió este modelo para el desarrollo de las pruebas en el escenario definitivo.

### 3.2.1.4 Edimax EW-7718Un



Figura 3-32. Edimax EW-7718Un

Se trata de un adaptador USB un poco más antiguo que los probados anteriormente, pero se recomendó su compra puesto que cumplía con los estándares necesarios y según el fabricante, el asistente de configuración permitía activar el funcionamiento de QoS y WMM. Sin embargo, como pudo comprobarse en las pruebas, no ofrecía dichas opciones ni registros asociados a la activación de las mismas.

Aún así se realizaron capturas similares a las realizadas para los demás adaptadores y se pudo comprobar que durante la fase de asociación el adaptador no indica la intención de utilizar WMM-PS para ninguna de las AC.





# 4 PRUEBAS SOBRE ESCENARIOS REALES

*Engineers like to solve problems. If there are no problems available, they will create their own problems.*

Scott Adams

**E**ste capítulo se dedica a la descripción de las pruebas realizadas para simular escenarios reales y a partir de las cuales se pueden obtener resultados significativos. Se analizarán los escenarios y dispositivos empleados para las pruebas así como el software empleado.

## 4.1 El escenario

En primer lugar es necesario conocer el escenario real que se ha pretendido simular; en la figura que se muestra a continuación se muestra un esquema simplificado del mismo.

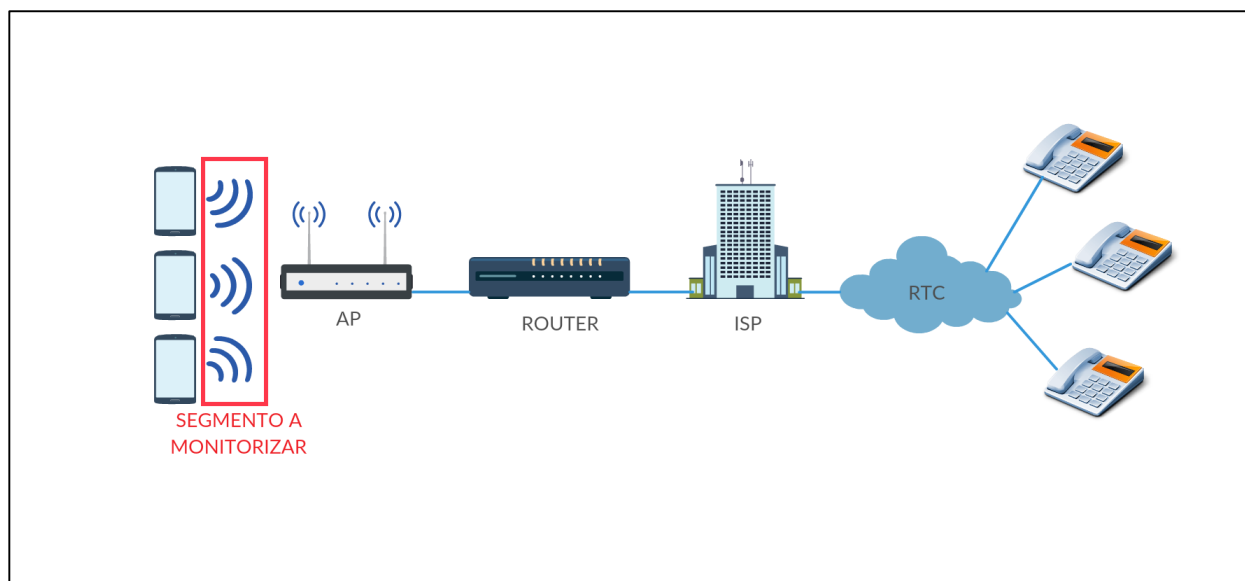


Figura 4-1. Esquema del escenario

Básicamente se desea recrear una red inalámbrica que de acceso a varios terminales móviles que realicen llamadas de VoIP hacia otros terminales que en este caso se encuentren cableados. Puesto que el segmento a monitorizar es el interfaz radio entre los dispositivos móviles y el AP, realmente no se entra en detalle con respecto a la señalización y el extremo cableado de la comunicación. En el esquema se ha supuesto que la comunicación es con teléfonos conectados a la Red Telefónica Conmutada, por lo que se hace necesaria una pasarela de medios y de señalización, aunque no se representa porque como se ha indicado no es el objeto principal de la investigación.

Una vez conocido el escenario a simular, se procede al análisis de la implementación escogida. En primer lugar hay que tener en cuenta varios aspectos:

- Se necesita coherencia entre los relojes de ambos extremos de la comunicación y del equipo que

actuará como sniffer, pues las marcas de tiempo permitirán calcular parámetros de calidad y el tiempo que el interfaz radio permanece en un estado u otro.

- Se precisa de un software para realizar las llamadas de VoIP que permita modificar los parámetros deseados para la batería de pruebas.
- Debido a la saturación de la banda de 2,4GHz será preferible el empleo de la banda de 5GHz para reducir las colisiones en el acceso al medio y el tamaño de los archivos de captura de paquetes generados por el sniffer.
- El intercambio de paquetes entre el AP y las estaciones puede implicar el empleo de RTS/CTS concretamente, para el adaptador de Intel empleado, es posible alternar entre los modos RTS-To-Self y RTS/CTS, por lo que se hicieron pruebas con ambos.

Teniendo en cuenta todas estas consideraciones, el escenario que se decidió utilizar fue el siguiente

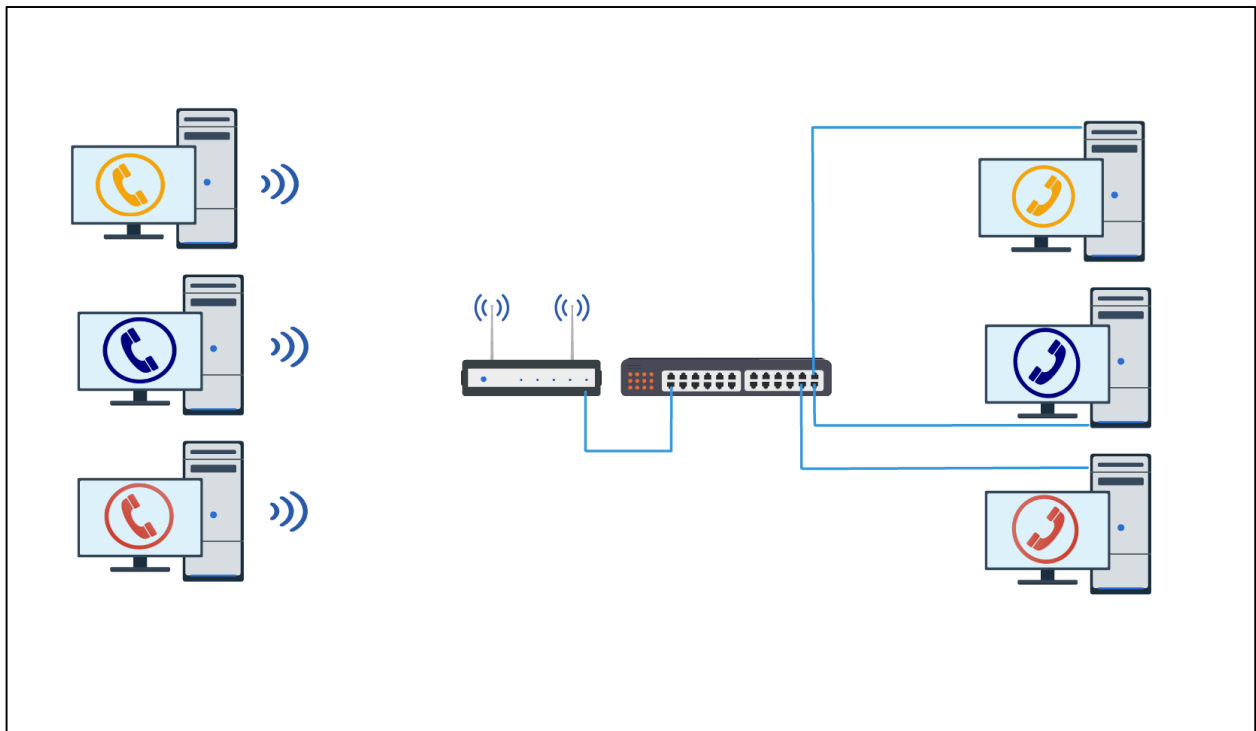


Figura 4-2. Escenario planteado originalmente

Es necesario señalar, que debido a las limitaciones de tiempo y equipos disponibles, se optó por emplear únicamente un par de equipos para el escenario entre los que se estableció la comunicación VoIP. De esta manera se monitorizará una única conversación entre un equipo con interfaz inalámbrico y un equipo con conexión Ethernet.

#### 4.1.1 Red de sincronismo

Para garantizar que los relojes de los equipos funcionen sincronizados se decidió utilizar el protocolo PTP (Precision Time Protocol), una implementación del estándar IEEE 1588. Este protocolo tiene un funcionamiento basado en el esquema maestro-esclavo por el cual un reloj maestro envía una serie de mensajes que indican el tiempo de su reloj y los esclavos se sincronizan con este tiempo.

##### 4.1.1.1 IEEE 1588

El estándar IEEE 1588 [26] está diseñado para sistemas cuyas especificaciones no puedan ser satisfechas de forma apropiada por cualquiera de los dos protocolos dominantes, NTP y GPS, como sistemas locales que requieran una precisión más allá de las alcanzables mediante NTP, y aplicaciones para las que un receptor de GPS en cada nodo resulte excesivamente costoso, o que no puedan acceder a las señales GPS.

En la figura que se muestra a continuación puede observarse como se realiza el procedimiento básico de sincronización utilizando varios mensajes definidos para el protocolo:

- Los mensajes *Sync*, *Delay\_Req*, *Follow\_Up* y *Delay\_Resp* son utilizados por los relojes ordinarios y maestros y permiten comunicar información temporal utilizada para sincronizar los relojes de la red.
- Los mensajes de anuncio son empleados por el *algoritmo de selección para el mejor reloj maestro* definido en la segunda revisión del protocolo para construir una jerarquía de relojes y seleccionar al mejor maestro.
- Los mensajes de gestión se emplean para monitorizar, configurar y mantener un sistema PTP completo.
- Los mensajes de señalización se emplean para las comunicaciones entre relojes que no son críticas en tiempo.

Los mensajes se categorizan como mensajes de eventos y generales. Los mensajes de eventos son críticos en tiempo ya que la precisión en su transmisión y recepción afecta directamente a la precisión de la distribución del tiempo del reloj maestro. Los mensajes *Sync*, *Delay\_Req*, y *Delay\_Resp* son mensajes de eventos. Por el contrario, en los mensajes generales, como los mensajes *Follow\_Up* la precisión no es tan importante.

En la primera versión de PTP (IEEE 1588-2002) todos los mensajes se envían utilizando direcciones multicast, mientras que en su segunda revisión (IEE 1588-2008, la empleada en las pruebas), se introduce una opción que permite a los dispositivos negociar la transmisión unicast sobre un esquema puerto a puerto. En concreto, la configuración empleada en el escenario utilizado empleaba anuncios del maestro en multicast, y comunicación con los esclavos en unicast. Dicha configuración aparece como se ha comentado previamente en el Anexo A.

Los mensajes PTP utilizan UDP como protocolo de transporte sobre IP. La versión original empleaba únicamente el transporte sobre IPv4, y con la segunda revisión se introduce el soporte de IPv6. Los datagramas se transmiten empleando el direccionamiento IP de multicast, con las direcciones de grupo definidas para IPv4 e IPv6. Los mensajes de evento utilizan el puerto 319, los generales el 320 y las respuestas a los mensajes de gestión se envían siempre a la dirección unicast del originador del mensaje. [31]

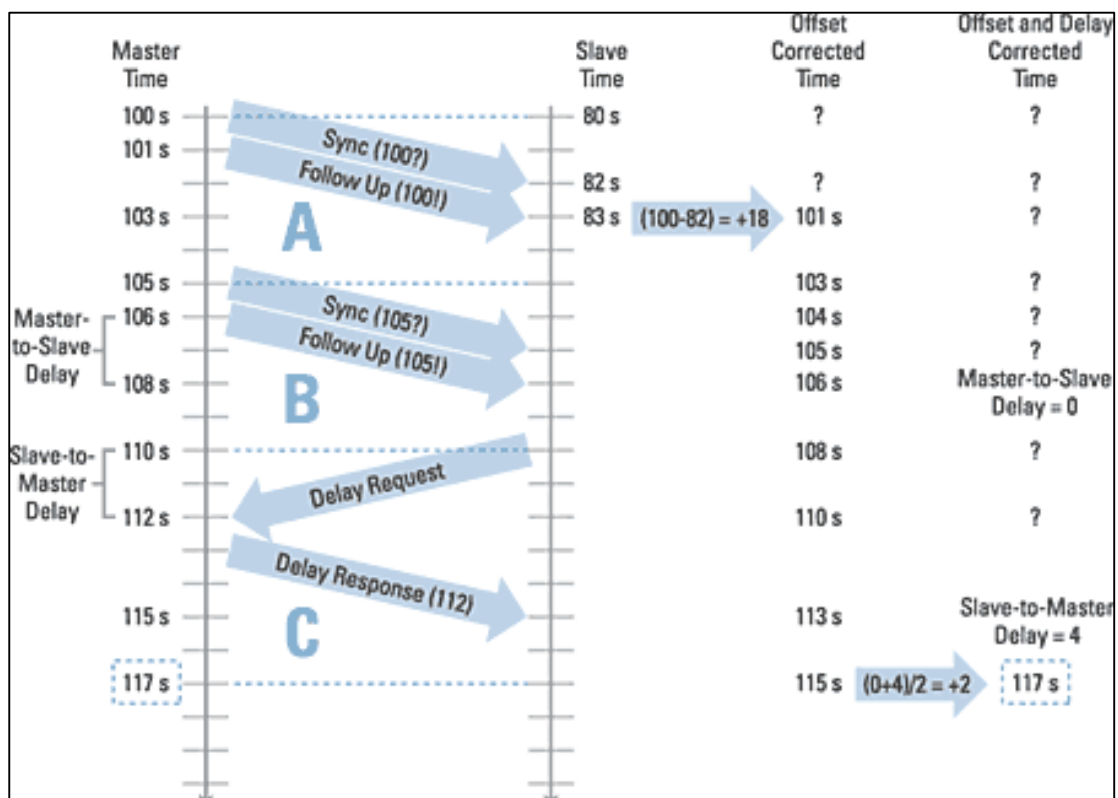


Figura 4-3. Esquema simplificado del intercambio de mensajes PTP [18]

Utilizando PTP, los equipos determinan el offset entre el maestro y su propio reloj. Sea  $t$  el tiempo físico. Para un determinado esclavo, el offset  $o(t)$  en el instante  $t$  se obtiene como:

$$o(t) = s(t) - m(t)$$

Siendo  $s(t)$  el tiempo medido por el esclavo y  $m(t)$  el tiempo por el maestro, ambos en el instante  $t$ .

El maestro difunde periódicamente el tiempo actual hacia los otros relojes. En la versión original, las difusiones podían estar separadas como mínimo por un periodo de 1 segundo, mientras que con la segunda revisión se pueden alcanzar hasta 10 difusiones por segundo.

Cada difusión del maestro comienza en un tiempo al que llamaremos  $T_1$  con el mensaje *Sync* que envía a todos los relojes de la red. Todo reloj que reciba esta mensaje, almacena tanto  $T_1$  como  $T_1'$ , el instante en el que el mensaje fue recibido. Posteriormente, el maestro puede enviar un mensaje *Follow\_Up* de multicast con el timestamp preciso de  $T_1$ . Esto ocurre cuando el maestro no tiene capacidad de enviar un timestamp preciso en el mensaje *Sync* inicial.

Con el objetivo de obtener una sincronización precisa, cada reloj debe determinar individualmente el tiempo de tránsito de los mensajes *Sync* sobre la red. Para determinarlo, se hace de forma indirecta, midiendo el Round-Trip Delay Time<sup>1</sup> (RTT) de cada reloj hacia el maestro. De esta forma, los equipos comienzan un intercambio con el maestro que permite medir el tiempo de tránsito  $d$ .

El intercambio comienza con el esclavo enviando un mensaje *Delay\_Req* en el instante  $T_2$  hacia el maestro. Cuando el maestro lo recibe en el instante  $T_2'$ , responde con un *Delay\_Resp* que incluye la marca de tiempo  $T_2'$ .

Siendo  $\emptyset$  el offset constante entre relojes esclavos y el maestro:

$$\begin{aligned} T_1' - T_1 &= \emptyset + d \\ T_2' - T_2 &= -\emptyset + d \end{aligned}$$

Combinando ambas ecuaciones:

$$\emptyset = (T_1' - T_1 - T_2' + T_2) / 2$$

De esta manera, cada equipo conoce el offset durante la transacción y puede corregir su reloj con esta cantidad para estar sincronizado con el maestro.

Para realizar estos cálculos, se ha asumido una serie de condiciones:

- El intercambio de mensajes ocurre en un periodo de tiempo tan pequeño que puede el offset puede ser considerado constante en un periodo.
- El tiempo de transito de un mensaje desde el maestro al esclavo es igual al tiempo de transito desde el esclavo al maestro.
- Tanto el maestro como el esclavo pueden realizar mediciones precisas del tiempo en el que envían o reciben un mensaje.

El grado en el que estas suposiciones se cumplan determinará la precisión del sincronismo. En una red de área local, empleando dispositivos de alta precisión, se alcanza una precisión de reloj en la gama de submicrosegundos, adecuado para los sistemas de medición y control.

En las pruebas realizadas y documentadas en el Anexo A se alcanzaron desviaciones de entre 0,411 y 0,219 ms, tiempos relativamente altos para el protocolo PTP. No obstante, es necesario tener en cuenta, que empleamos un switch que no está diseñado para el soporte de protocolos de tiempo de alta precisión y el reloj maestro tampoco es un reloj de alta precisión, sino el reloj de un equipo, por lo que consideraremos estos resultados aceptables.

<sup>1</sup> Se define como el tiempo que tarda un paquete de datos enviado desde un emisor en volver al mismo emisor habiendo pasado por el receptor de destino.

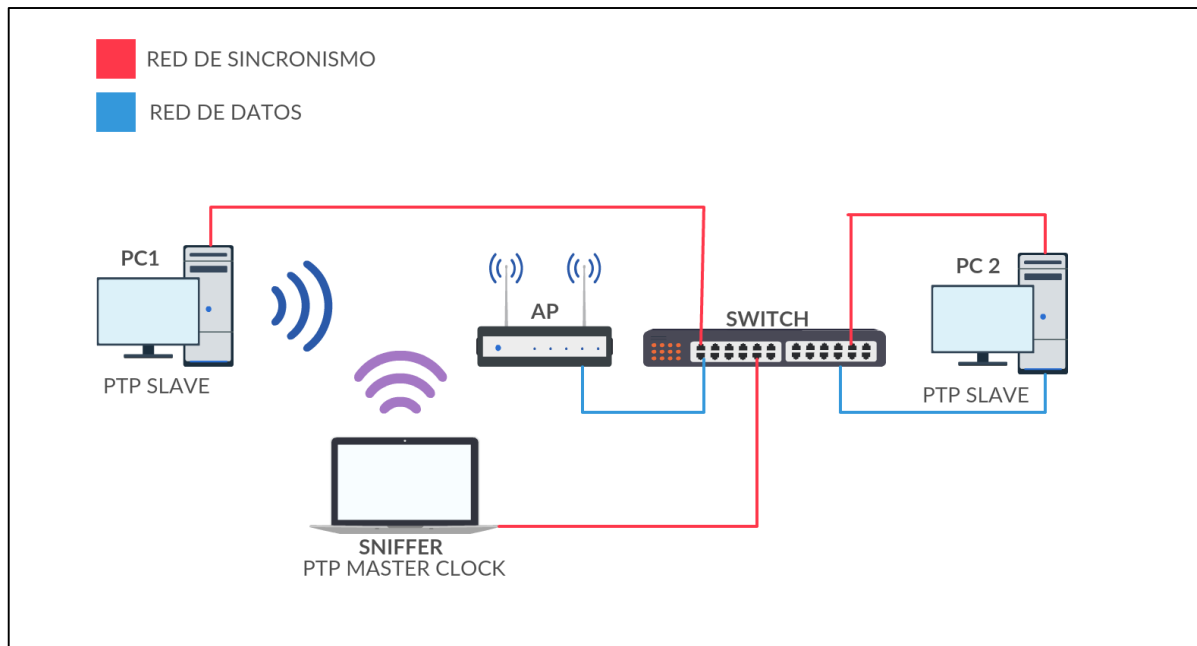


Figura 4-4. Esquema de la red utilizada para las pruebas

Para la red de sincronismo se utilizó el prefijo 10.2.0.0/24 y el software Domain Time II en los equipos finales y el demonio ptpdv2 en el equipo sniffer que actuó como reloj maestro. Los pasos para llevar a cabo la instalación del software y su configuración se recogen en el Anexo A.

Como puede observarse en la figura 4-4, y siguiendo las recomendaciones de la norma IEEE 1588 la red de sincronismo es totalmente cableada, para evitar que los mecanismos de acceso al medio inalámbrico puedan distorsionar las medidas de la desviación de los relojes y evitar un correcto sincronismo. [19] [20]

#### 4.1.2 Red de datos

Para evitar que el tráfico del protocolo PTP se mezclara con el de voz y pudiera alterarse el resultado de las pruebas, se optó por utilizar una red diferente para VoIP, con el prefijo 10.1.0.0/24. Para la división de la red en dos redes lógicas (como se observa en la figura 4-4) se emplearon dos VLANs diferentes en el switch. Los pasos seguidos para llevar a cabo esta configuración pueden encontrarse en el Anexo A.

Los parámetros más relevantes del AP son:

- 802.11n
- Banda de frecuencia de 5GHz
- Ancho de banda canal automático entre 20 MHz y 40 MHz
- Beacon interval 500ms
- Ack timeout 25  $\mu$ s
- Tasa 54 Mb/s

## 4.2 PJSIP

El software que se empleó para realizar y recibir llamadas de VoIP en los equipos finales fue PJSIP.

PJSIP es un software de VoIP de código abierto y gratuito que consiste en un conjunto de librerías multimedia de comunicación y utilidades VoIP que implementan protocolos basados en estándares como SIP, SDP, RTP, STUN, TURN, y el ICE. Combina el protocolo de señalización, SIP, con el manejo de flujos multimedia y funcionalidades NAT traversal. Además, es portátil y adecuado para casi cualquier tipo de sistemas que van desde equipos de escritorio, sistemas embebidos o teléfonos móviles. El código fuente PJSIP se encuentra

disponible en el directorio `/Software/pjproject-2.5.1` del CD adjunto.

PJSIP trata de proporcionar a los desarrolladores todo lo necesario para construir aplicaciones de comunicación multimedia en tiempo real en lenguaje de programación C. Puede ser ejecutado prácticamente desde cualquier tipo de sistema, como Windows, Linux o sistemas Unix entre otros, incluyendo también sistemas operativos para smartphones.

La estructura de PJSIP consiste en múltiples niveles de APIs; cada una de estas capas se encuentra una encima de otra y relacionadas entre sí. La más básica de todas las capas (PJLIB) comienza definiendo sus propios tipos de datos y estructuras.

PJSIP incluye un potentísimo cliente SIP en línea de comandos llamado PJSUA (véase figura 4-4) que nos permite realizar llamadas a través de la red. PJSUA es una API de alto nivel para la construcción de agentes de usuario SIP multimedia. Engloba señalización y funciones multimedia en una aplicación fácil de usar. Sus principales características son:

- Administración de múltiples cuentas
- Administración de amigos
- Mensajería instantánea
- Múltiples llamadas
- Retención y transferencia de llamadas
- Conferencias
- Auto-responder con un fichero de audio (lo cual permitió automatizar las pruebas)
- Grabación de voz (empleada para la evaluación de QoS)

```

Simbolo del sistema - pjsua-mod.exe --config-file=config.txt --null-audio
+-----+-----+-----+
| Call Commands: | Buddy, IM & Presence: | Account: |
+-----+-----+-----+
| m Make new call | +b Add new buddy      | +a Add new acct |
| M Make multiple calls | -b Delete buddy      | -a Delete acct. |
| a Answer call    | i Send IM            | !a Modify acct. |
| h Hangup call (ha=all) | s Subscribe presence | rr (Re-)register |
| H Hold call     | u Unsubscribe presence | ru Unregister   |
| v re-inVite (release hold) | t ToGgle Online status | > Cycle next ac. |
| U send UPDATE   | T Set online status  | < Cycle prev ac. |
+-----+-----+-----+
| j, l Select next/prev call | | |
| x Xfer call      | | |
| X Xfer with Replaces | | |
| # Send RFC 2833 DIMF | | |
| * Send DIMF with INFO | | |
| dq Dump curr. call quality | | |
| S Send arbitrary REQUEST | | |
+-----+-----+-----+
| Media Commands: | Status & Config: |
+-----+-----+-----+
| cl List ports    | d Dump status     |
| cc Connect port | dd Dump detailed  |
| cd Disconnect port | dc Dump config    |
| U Adjust audio Volume | f Save config     |
| Cp Codec priorities | | |
+-----+-----+-----+
| q QUIT L ReLoad sleep MS echo [0|1|txt] n: detect NAT type |
+-----+-----+-----+
You have 0 active call
>>>

```

Figura 4-5. Consola de comandos de PJSUA

Sobre la versión original de PJSUA se han realizado algunas modificaciones para facilitar la gestión y automatización de las pruebas que se recogen en el Anexo B. Para detalles a cerca de su instalación y configuración puede consultarse el Anexo A.

A grandes rasgos, la modificación principal que se introdujo en el código permite indicar el número de tramas por paquetes a utilizar para cada ejecución del programa mediante un fichero de texto externo con el nombre `NFPP.TXT`. De esta forma, se puede modificar de forma sencilla el número de tramas por paquete empleadas por cada códec y realizar así distintas pruebas.

Es necesario señalar que no se utilizaron todas las modificaciones sobre el software que se introdujeron en el proyecto anterior, que pueden encontrarse en la carpeta `/Software/proyectoPJSIP/modificaciones` del

CD adjunto por problemas de compatibilidad entre las versiones del código fuente de pjsua. Estas modificaciones básicamente permitían recoger estadísticas a cerca de la evolución del Factor R, RTT y retardo.

### 4.3 Pruebas

El objetivo principal durante el desarrollo del trabajo fue utilizar el mayor número de códecs posibles para comparar las estadísticas obtenidas, pero debido a restricciones del software y problemas de compatibilidad que analizaremos, no todos pudieron ser probados.

Las conversaciones empleadas proceden de un banco de conversaciones adquirido por el departamento de telemática: *Linguistic Data Consortium* de la Universidad de Pennsylvania [34]. Para cada conversación (2 extremos) de 15 minutos, se planteó inicialmente la siguiente batería de pruebas:

- Nfpp=1:10 (x10)
- Codec=[g711,Opus,iLBC, AMR-NB, AMR-WB,G729A] (x6)
- PSM=[noPSM, PSM, UAPSD] (x3)

Tras realizar una serie de pruebas iniciales, se pudo determinar que el comportamiento de el adaptador de red en el modo PSM tradicional es igual que el funcionamiento con PSM desactivado, ya que a pesar de tener la posibilidad de dormir en intervalos de Beacon, cuando se generó el tráfico de VoIP se pudo comprobar que el adaptador no indicaba que entrase en el modo PSM ni tampoco solicitaba las posibles tramas almacenadas. Por tanto, para las pruebas se utilizaron dos casos en lo que a modos ahorro de energía se refiere: UAPSD, noUAPSD.

Respecto a los códecs, finalmente se pudieron utilizar exclusivamente G711(PCMA), iLBC y Speex (predecesor de Opus), ya que Opus se ofrece exclusivamente para las implementaciones en plataformas móviles. Por otra parte, tanto AMR como G729 pueden incorporarse al software empleando las *Intel Integrated Performance Primitives* [29], no obstante, las librerías que se requieren son de pago.

Se intentó integrar las librerías a través de las versiones gratuitas de *Intel Parallel Studio xe* [30]. Estas versiones de evaluación pudieron ser descargadas tras registrar un perfil de estudiante en la propia web de Intel [31]. No obstante, las librerías relativas a dichos códecs no se encontraban incorporadas en el conjunto de librerías, por lo que no se pudieron integrar finalmente con el software pjsua.

Finalmente, y teniendo en cuenta las restricciones previamente planteadas, la batería de pruebas realizadas fue la siguiente:

- Nfpp=1:10 (x10 salvo iLBC que falla a partir de Nfpp=8)
- Codec=[PCMA, iLBC, Speex] (x3)
- PSM=[noUAPSD, UAPSD] (x2)
- Protección de modo mixto=[RTS/CTS, RTS-to-Self] (x2 sólo con PCMA)

Los parámetros específicos empleados para cada códec se recogen a continuación:

Tabla 4-1. Comparativa de los códecs empleados

| Nombre | Duración de trama (ms) | Frecuencia de Muestreo (kHz) | Bitrate (kb/s) | VAD | Nfpp Máximo (1-10) |
|--------|------------------------|------------------------------|----------------|-----|--------------------|
| Speex  | 30                     | 8                            | 2.15-24.6      | Si  | 10                 |

|                 |                             |   |       |    |    |
|-----------------|-----------------------------|---|-------|----|----|
| G.711<br>(PCMA) | Dependiente<br>del muestreo | 8 | 64    | Si | 10 |
| iLBC            | 30                          | 8 | 13.33 | Si | 7  |

### 4.3.1 Terminales VoIP

Con el objetivo de automatizar las pruebas y aprovechar la posibilidad de modificar el número de tramas por paquete se codificó así mismo un script de Windows PowerShell capaz de ejecutar una batería de pruebas. El script, que puede encontrarse en la carpeta `/Software/config/PCX/pjsip_g.ps1` del CD adjunto va lanzando llamadas de forma consecutiva a la vez que incrementa el Nfpp para cada nueva llamada. PCX indica el pc en el que se ha ejecutado, ya que hay ligeras variaciones en función de si se trata del equipo llamante o del que recibe la llamada. Se puede encontrar un análisis más detallado del código del script en el Anexo B.

Para que dicho script funcione correctamente, es necesario colocarlo en el directorio que se desee y acompañarlo de la carpeta `NFPPS` que puede encontrarse en `/Software/config/PCX/` del CD, junto con la configuración que se desee utilizar (`config_pcma.txt`, `config_ilbc.txt`, `config_speex.txt`) que se puede encontrar en el mismo directorio del CD y con el ejecutable de pjsua compilado; `pjsua-mod.exe`.

Cuando se desee cambiar el códec a utilizar, es tan sencillo como modificar el nombre el fichero `.txt` que se pasa como parámetro en la llamada al ejecutable tras `--config=` dentro del script.

En la siguiente figura se observa un fragmento de uno de los ficheros de configuración que se proporcionan, concretamente `config_speex.txt`. En este fragmento pueden observarse alguna de las opciones más relevantes que ofrece pjsua, como por ejemplo la posibilidad de desactivar codecs, para evitar su anuncio en la negociación con SDP, el fichero de audio a utilizar como fuente de voz para la llamada, o la duración total de la misma.

```

--add-codec=speex/8000
--dis-codec=pcma
--dis-codec=pcmu
--dis-codec=ilbc
--dis-codec=gsm
--dis-codec=g722
--play-file=Conv_1_ext_1.wav
--snd-auto-close=0
--auto-play
--auto-conf
--auto-answer=200
--duration=900

```

Figura 4-6. Ejemplo de fichero de configuración

Para modificar el modo de ahorro de energía, basta con seguir los pasos explicados en el capítulo 3 o su versión más esquemática recogida en el Anexo A.

Es necesario señalar que el script de Windows PowerShell genera en el directorio en el que se ejecuta una serie de grabaciones, que contienen el audio recibido en cada extremo de la conversación, para poder realizar un análisis de la calidad de audio recibida (se explica a continuación). Estas grabaciones generadas, toman el nombre `OUTPUTX.WAV` siendo X el valor de Nfpp en la prueba concreta.



Concluyendo, para lanzar una batería de pruebas, basta con ejecutar de forma simultánea el script tanto en el equipo llamante como en el equipo llamado, tras haber comprobado que los relojes de ambos están correctamente sincronizados con el del maestro.

### 4.3.2 Sniffer

En el equipo encargado de capturar los paquetes (Macbook Pro Retina 13 con OSX El Capitán) existen varias posibilidades. Puede utilizarse *Wireshark* o su versión por línea de comandos *tshark*, ambas en modo monitor (cabecera Radiotap). Por comodidad, y la posibilidad de ejecutarlo en segundo plano, se ha utilizado normalmente *tshark* para la realización de las capturas.

Para realizar una captura con *tshark*, basta con ejecutar en un terminal la siguiente línea:

```
tshark -i enX -b duration:B -a duration:A -w FICHERO.pcap -F pcap -I -f "FILTRO"
```

donde *enX* indica el interfaz en el que se desean capturar los paquetes, *B* la duración máxima de cada fichero de captura, *A* el tiempo máximo durante el que *tshark* estará capturando en el interfaz correspondiente, *FICHERO.pcap* el fichero de captura en el que se almacenarán los paquetes capturados, y la expresión "FILTRO" indica el filtro o filtros a utilizar siguiendo la sintaxis de la librería *pcap* [32].

Para facilitar las capturas se ofrece también un script que se encarga tanto de realizar las capturas como de clasificarlas y comprimirlas, aplicando diferentes filtros. Originalmente estaba ideado para realizar capturas en un escenario con varias llamadas simultáneas, por lo que puede ser muy útil para seguir desarrollando el proyecto en un futuro. Puede encontrarse en */Software/autocapture.sh* del CD adjunto.

# 5 ANÁLISIS DE LOS RESULTADOS

*An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.*

John W. Tukey

**E**ste capítulo se centrará en el análisis de los resultados obtenidos tras la realización de las pruebas, así como del procesamiento de los mismos, con el objetivo de obtener estadísticas y parámetros que nos permitan determinar los beneficios del empleo del mecanismo de ahorro de energía y su relación con los diferentes parámetros y códecs que se han variado en las pruebas.

## 5.1 Resultados de las pruebas

Como se comentó en el capítulo anterior, cada llamada de 15 minutos, generó un fichero de audio de extensión .WAV en cada extremo de la conversación en el que se almacena la llamada recibida. Así mismo se obtiene un fichero de captura .pcap por cada llamada de 15 minutos, resultante de la monitorización del interfaz radio entre el AP y la estación.

Los ficheros de audio serán empleados para obtener estimaciones subjetivas de la calidad de servicio percibida, mientras que los ficheros pcap permitirán analizar el intercambio de mensajes entre la estación y el AP con el objetivo de determinar el consumo energético a partir de el tiempo que el adaptador permanece en cada estado (transmisión, recepción, ocioso o dormido) y estadísticas para la evaluación de la calidad de servicio en función del número de tramas por paquete.

No obstante, no es posible actuar sobre el fichero .pcap directamente para obtener estos resultados, por lo que se volcó su contenido en ficheros de texto plano, utilizando la opción de Wireshark *File>Export Packet Dissections>As Plain Text* (es importante no marcar la casilla *Details* y marcar la casilla *Bytes* antes de exportar). Sobre este fichero actúa un programa desarrollado para obtener la estadísticas.

### 5.1.1 Software para el análisis: UAPSD

Como se ha comentado, es esencial el procesamiento de los datos obtenidos tras la realización de las pruebas partiendo de un fichero de texto plano. A diferencia de el trabajo de fin de carrera del que parte este proyecto, no se ha optado por usar *awk* para su procesamiento, y esto se debe a que el cómputo del tiempo que pasa la estación en cada estado no puede obtenerse como un valor medio de los datos de los que se dispone, sino que es necesario recorrer el fichero de forma secuencial, conociendo los paquetes previamente recibidos y transmitidos y utilizando una serie de acumuladores y variables bandera que nos permitan determinar el estado en el que se encuentra.

Inicialmente se utilizó un script de *bash* para el procesamiento de los ficheros de texto, por su versatilidad en el manejo de las cadenas de caracteres, pero tras realizar varias pruebas se pudo comprobar que se trataba de un lenguaje muy poco eficiente para dicho propósito, pues el tiempo que requería para el procesamiento de cada línea y las operaciones encadenadas a realizar, era demasiado grande. Un fichero de captura ocupa entre 1000000 y 2000000 líneas de un fichero de texto plano, y se pudo comprobar que el tiempo aproximado en el que se procesaban 250000 líneas (1/4 del tamaño aproximado) se encontraba cercano a una hora, por lo que finalmente se abandonó esta idea.

Por tanto se hizo necesario utilizar un lenguaje de programación alternativo, concluyendo que *C* sería una

solución adecuada. Es necesario señalar que aunque la gestión de los ficheros y concretamente las cadenas de texto y bytes de las capturas es más compleja en este lenguaje, el rendimiento es muy superior al ofrecido por bash; se puede procesar un fichero en dos o tres segundos.

Una vez encontrado el lenguaje a utilizar, se hizo necesario plantear la manera en la que el programa recorrería cada fichero y calcularía el tiempo que el adaptador pasaba en cada estado. Para ello, se analizaron con detenimiento las diferentes capturas de Wireshark, buscando patrones repetitivos que permitiesen predecir el funcionamiento de la estación y el AP.

Tras este análisis comparativo entre las diferentes capturas y en el que también se tuvo en cuenta el funcionamiento de U-APSD establecido en la norma 802.11e, se pudieron identificar una serie de patrones repetitivos, una versión simplificada del funcionamiento que se concibió de forma similar a una máquina de estados en la que se puede pasar de unos a otros.

Se habla de una versión ‘simplificada’, porque en ella no se tienen en cuenta algunos aspectos que si se gestionan y comprueban en el programa, como por ejemplo el tráfico ARP entre las estaciones y los equipos cableados, algunos mensajes de gestión de LLC/SNAP, paquetes fragmentados, retransmisiones y paquetes que Wireshark captura de forma errónea entre otros.

Todos estos aspectos tienen indudablemente una repercusión sobre la secuencia en la que se produce el intercambio de mensajes del mecanismo de ahorro de energía, pero su ocurrencia es mucho menor que la de los paquetes de RTP, por ello, no se tienen en cuenta en el modelo de estados que se muestra a continuación, pero si en el programa. Un análisis más detallado del mismo, en el que se especifican todos estos casos puede encontrarse en el Anexo B.

#### **5.1.1.1 Algoritmo simplificado**

A continuación se analizan con detalle los diferentes estados considerados, así como las transiciones entre los mismos y el intercambio de mensajes que se produce dentro de cada uno de ellos. Nótese que a la izquierda se muestra el patrón energético que tiene lugar en cada estado y que se empleará para el cálculo de la energía consumida.

Hay una consideración importante que se debe señalar: el AP utiliza el mecanismo RTS/CTS antes de transmitir información a la estación con independencia de que esta lo utilice o no; de hecho, como se comentó anteriormente, en la estación dicho mecanismo se pudo desactivar para la realización de las pruebas, pero en el AP no fue posible hacerlo, ya que no ofrecía ninguna opción que permitiese deshabilitar su funcionamiento. Esto implica una complejidad adicional en el cómputo de los estados pero a su vez genera un patrón reconocible que facilitó el diseño del algoritmo.

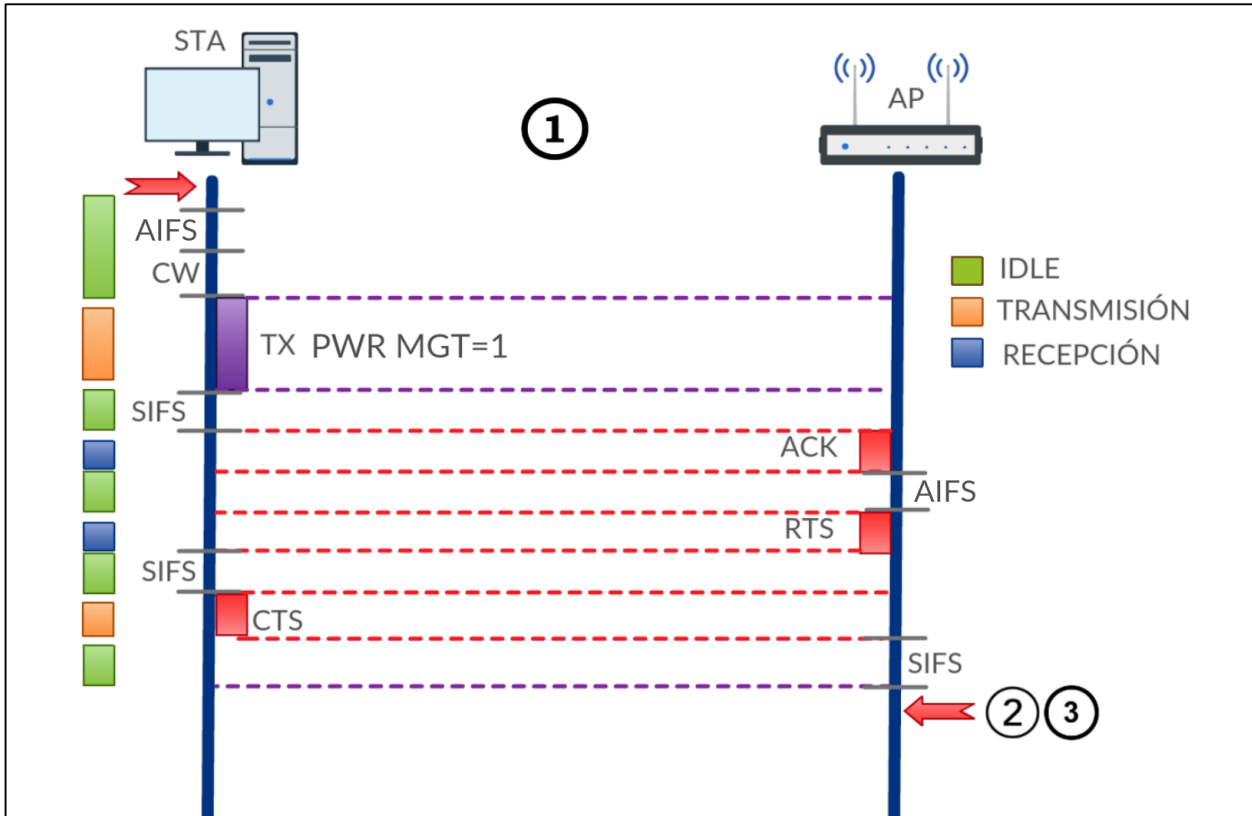


Figura 5-1. Primer estado

El primer estado comienza con el despertar de la estación, en este caso porque tiene datos que transmitir. La estación espera en primer lugar un periodo AIFS en lugar de DIFS por ser EDCA el modo de acceso al medio. El valor de AIFS dependerá de la categoría de acceso, en este caso AC\_VO, y su valor se obtiene como:

$$\text{AIFSN}[\text{AC}] * \text{TS} + \text{SIFS}$$

donde Ts indica la duración del time slot, que depende del medio físico, y que para 802.11n en la banda de 5Ghz es de 9µs, AIFSN es un número que depende de la categoría de acceso y que fija el AP (puede consultarse en las tramas Beacon y de asociación), y SIFS toma el valor de 16µs.

- ▶ Ac Parameters ACI 0 (Best Effort), ACM no, AIFSN 3, ECWmin 4, ECWmax 10, TXOP 0
- ▶ Ac Parameters ACI 1 (Background), ACM no, AIFSN 7, ECWmin 4, ECWmax 10, TXOP 0
- ▶ Ac Parameters ACI 2 (Video), ACM no, AIFSN 2, ECWmin 3, ECWmax 4, TXOP 94
- ▶ Ac Parameters ACI 3 (Voice), ACM no, AIFSN 2, ECWmin 2, ECWmax 3, TXOP 47

Figura 5-2. Detalle de un beacon del AP: valores de los parámetros de EDCA

Una vez que ha esperado el AIFS correspondiente, espera el número de time slots que haya obtenido aleatoriamente. Los valores ECWmin y ECWmax permiten determinar el tamaño máximo y mínimo de la ventana de contención empleada.

Si la estación gana el canal transmite un paquete de la categoría de voz en este caso, con el bit PWR MGT a uno solicitando de esta forma cualquier paquete que pudiera encontrarse almacenado en el buffer de la AC de voz del AP.

Una vez que el AP lo recibe, responde con un ACK y comienza el mecanismo de RTS/CTS previo al envío del paquete de respuesta. Así finaliza este primer estado, del que se puede pasar a dos estados que son bastante similares, y cuya diferencia es que en uno de ellos el AP tiene más de un paquete que enviar.

No obstante, es necesario señalar que la estación puede enviar varios paquetes durante su TXOP, por lo que a este primer estado “simplificado” habría que añadirle otro posible caso (cuestión que se tuvo en cuenta en el desarrollo del programa, creando un sexto estado).

|        |            |                            |  |        |                                      |
|--------|------------|----------------------------|--|--------|--------------------------------------|
| 257152 | 491.136... | 10.1.0.1                   | 10.1.0.2                                   | RTP    | PT=ITU-T G.711 PCMA, SSRC=0x294823,  |
| 257153 | 491.136... |                            | D-LinkIn_0a:5d:88 (c8:be:19:0a:5d:88) (RA) | 802.11 | Acknowledgement, Flags=.....         |
| 257154 | 491.137... | 10.1.0.2                   | 10.1.0.1                                   | RTP    | PT=ITU-T G.711 PCMA, SSRC=0x43451CEA |
| 257155 | 491.137... |                            | IntelCor_c8:c2:88 (7c:5c:f8:c8:c2:88) (RA) | 802.11 | Acknowledgement, Flags=.....         |
| 257156 | 491.137... | 10.1.0.2                   | 10.1.0.1                                   | RTP    | PT=ITU-T G.711 PCMA, SSRC=0x43451CEA |
| 257157 | 491.137... |                            | IntelCor_c8:c2:88 (7c:5c:f8:c8:c2:88) (RA) | 802.11 | Acknowledgement, Flags=.....         |
| 257158 | 491.156... | D-LinkIn_0a:5d:88 (c8:b... | IntelCor_c8:c2:88 (7c:5c:f8:c8:c2:88) (RA) | 802.11 | Request-to-send, Flags=.....         |
| 257159 | 491.156... |                            | D-LinkIn_0a:5d:88 (c8:be:19:0a:5d:88) (RA) | 802.11 | Clear-to-send, Flags=.....           |
| 257160 | 491.156... | 10.1.0.1                   | 10.1.0.2                                   | RTP    | PT=ITU-T G.711 PCMA, SSRC=0x294823,  |

Figura 5-3. Intercambio de mensajes con varios paquetes ascendentes en TXOP

Básicamente, la estación envía dos o más paquetes, recibiendo asentimientos a cada uno de ellos, o bien un asentimiento en bloque y posteriormente, comienza un ciclo de RTS/CTS por parte del AP al igual que en el estado 1 original.

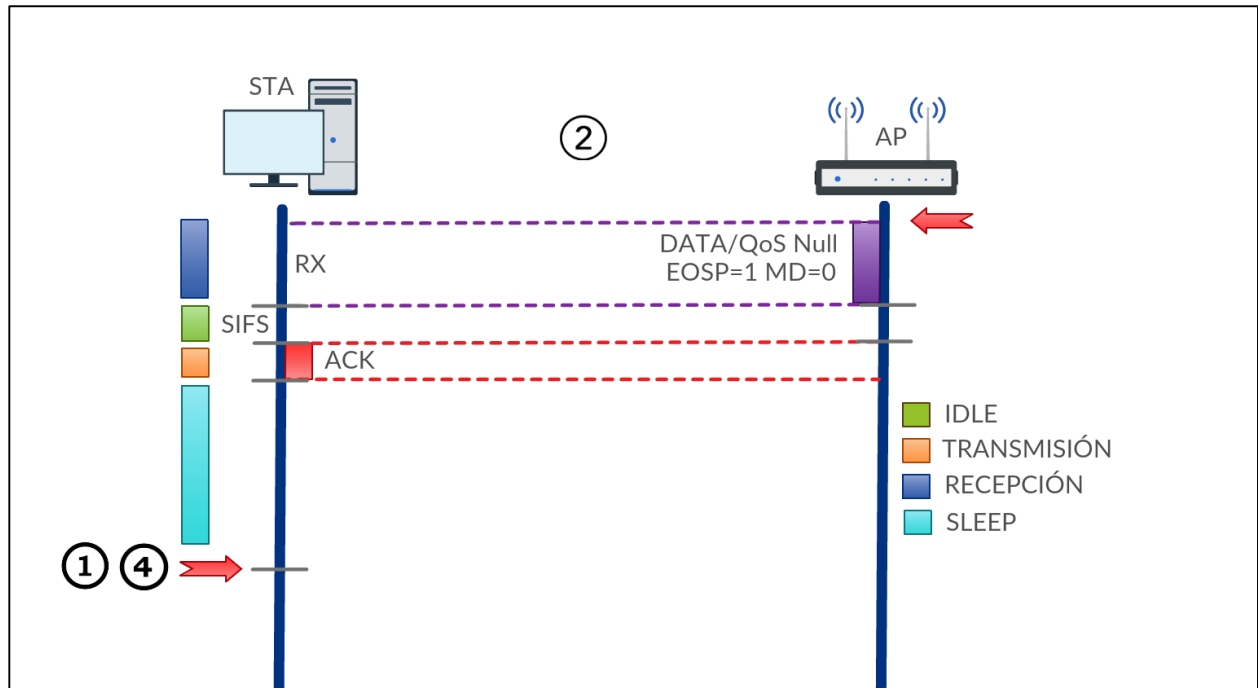


Figura 5-4. Segundo estado

En este segundo estado, el AP tras haber realizado previamente el mecanismo de RTS/CTS envía un paquete con los bits EOSP a uno y MD a cero, indicando que se acaba el periodo de servicio y que no hay más datos, o bien una trama QoS Null con los mismos flags si no dispone de datos que transmitir en el buffer de la AC que previamente se ha solicitado.

Una vez recibido el paquete, la estación lo asiente con el ACK y puede comenzar a dormir, bien hasta que se despierte porque tenga algo que transmitir (primer estado), o bien hasta que el algoritmo que controla el funcionamiento del adaptador decida que se debe despertar (cuarto estado). En el programa se toman medidas para determinar el tiempo medio y máximo de sueño, así como el periodo medio que el algoritmo utiliza para despertar a la estación cuando no tiene nada que transmitir. También existe la posibilidad de que se pase al quinto estado si al despertar la trama a transmitir es un informe RTCP.

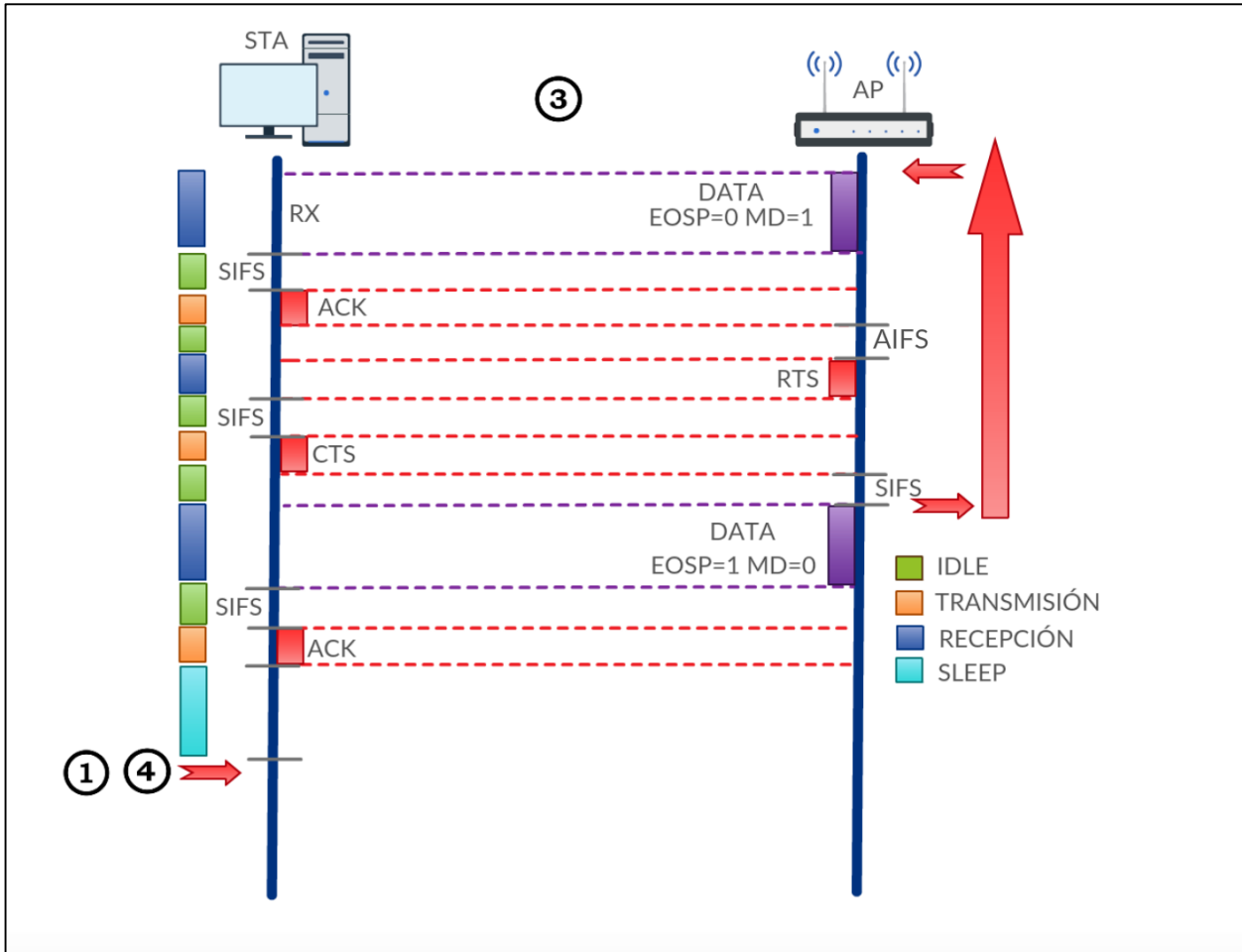


Figura 5-5. Tercer estado

Como comentamos anteriormente, este estado es similar al anterior, aunque la diferencia es que el AP puede tener varias tramas almacenadas en su buffer, lo cual se indica mediante el bit MD en este caso con valor 1.

Nótese que no acaba el periodo de servicio (EOSP=0) por lo que la estación sigue a la espera de recibir más información. Este comportamiento observado no es exactamente igual al definido en el estándar, según el cual, la estación debería enviar otro paquete o una trama QoS null solicitando más información en lugar de quedar a la espera.

Una posible explicación podría ser que el bit PWR MGT del ACK tuviera el valor 1, y efectivamente así es, aunque se pudo comprobar que la estación lo pone a 1 en todos sus ACK, por lo que se concluyó que la implementación del adaptador sencillamente espera a que el AP envíe la siguiente trama, en este caso, también con reserva previa.

Este proceso se podría repetir varias veces mientras no se exceda el TXOP del periodo de servicio, aunque se ha podido observar con el programa, que en la mayor parte de los casos se envían solo dos tramas. En el momento que se terminan las tramas del buffer o se excede el TXOP se envía la última trama con los bits EOSP a uno y MD a cero o a uno según corresponda.

La estación se dormirá en este momento, salvo que haya más datos en el buffer (MD=1), situación en la que intentará volver a ganar el canal, lo que la llevará nuevamente al primer estado o al cuarto (en función de si tiene datos para transmitir). En el caso de dormir, al despertar pasará también al primer estado o al cuarto. También existe la posibilidad de que se pase al quinto estado si al despertar la trama a transmitir es un informe RTCP.

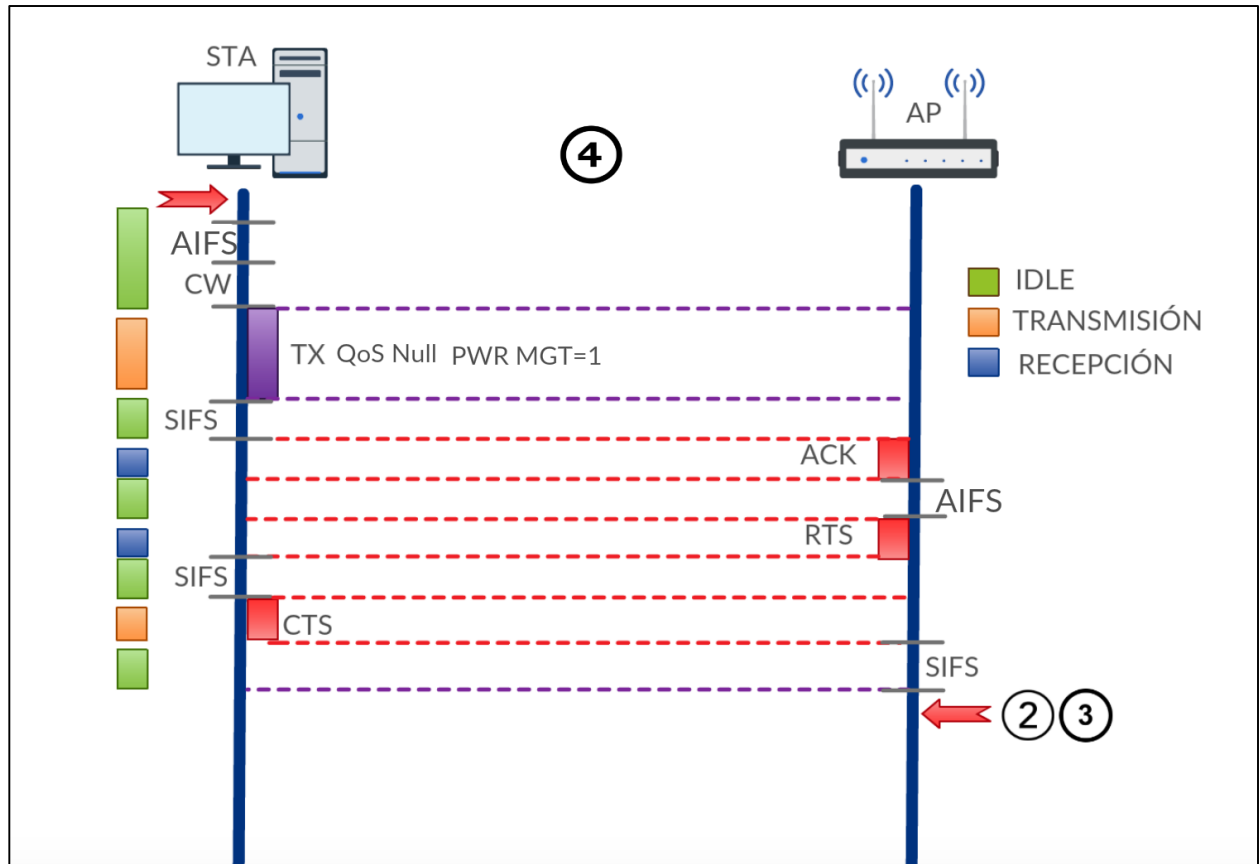


Figura 5-6. Cuarto estado

Este cuarto estado es muy similar al primero, la única diferencia es que a estación se despierta sin tramas que transmitir, y envía una trama QoS Null solicitando la información almacenada en el AP. Posteriormente el AP le envía el ACK correspondiente y comienza la reserva. Al igual que en el primer estado, puede pasarse directamente al segundo o al tercero.

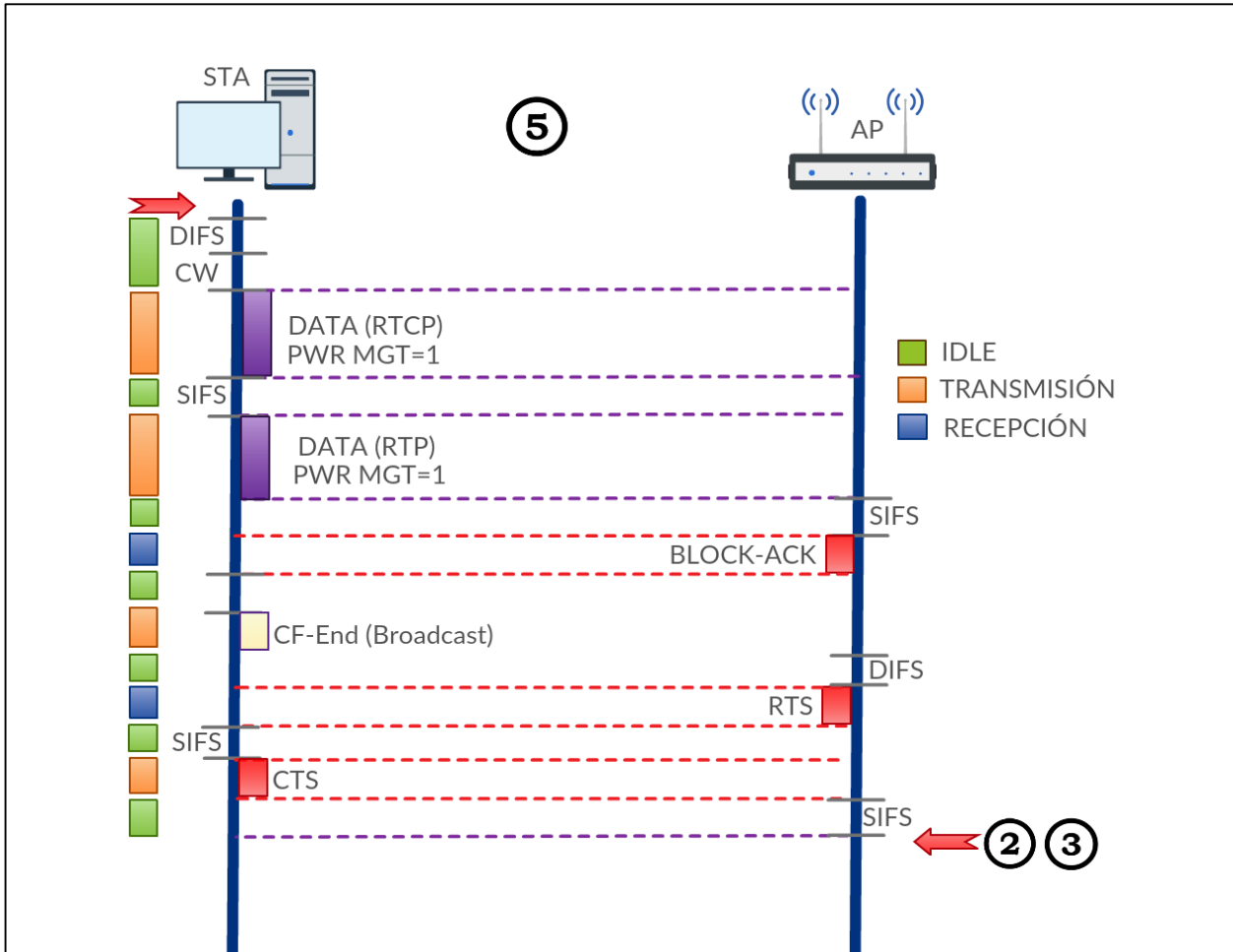


Figura 5-7. Quinto estado

Este quinto estado tiene lugar cuando la estación se despierta para transmitir un informe RTCP. En las capturas de Wireshark se pudo observar como siempre que aparece un informe de RTCP generado por la estación se opera de la misma forma.

Un vez enviado el primer paquete con el informe RTCP se envía un segundo paquete, en este caso con datos de RTP y posteriormente se emplea un asentimiento en bloque para ambos paquetes. Posteriormente, la estación transmite una trama especial de difusión (Contention Free End) y comienza un nuevo mecanismo de RTS/CTS por parte del AP. Al finalizar este estado se puede pasar tanto al segundo como al tercer estado.

Puede observarse este comportamiento en la siguiente captura de Wireshark, donde se pasa al estado dos (el AP envía una QoS Null):

|      |           |                            |  |        |   |
|------|-----------|----------------------------|--|--------|---|
| 4318 | 36.489538 | CiscoInc_b1:6a:7c          | Broadcast                                  | 802.11 | Beacon frame, SN=1843, FN=0, Flags=     |
| 4319 | 36.532009 | 10.1.0.2                   | 10.1.0.1                                   | 802.11 | RTCP Sender Report Source description   |
| 4320 | 36.532033 | 10.1.0.2                   | 10.1.0.1                                   | 802.11 | RTP PT=ITU-T G.711 PCMA, SSRC=0x5042054 |
| 4321 | 36.532040 | D-LinkIn_0a:5d:88 (c8:b... | IntelCor_c8:c2:88 (7c:5c:f8:c8:c2:88) (RA) | 802.11 | 802.11 Block Ack, Flags=.....           |
| 4322 | 36.532044 | IntelCor_c8:c2:88 (7c:5... | Broadcast (ff:ff:ff:ff:ff:ff) (RA)         | 802.11 | CF-End (Control-frame), Flags=...P.     |
| 4323 | 36.532133 | D-LinkIn_0a:5d:88 (c8:b... | IntelCor_c8:c2:88 (7c:5c:f8:c8:c2:88) (RA) | 802.11 | Request-to-send, Flags=.....            |
| 4324 | 36.532137 | D-LinkIn_0a:5d:88 (c8:b... | D-LinkIn_0a:5d:88 (c8:be:19:0a:5d:88) (RA) | 802.11 | Clear-to-send, Flags=.....              |
| 4325 | 36.532219 | D-LinkIn_0a:5d:88          | IntelCor_c8:c2:88                          | 802.11 | QoS Null function (No data), SN=348     |

Figura 5-8. Patrón del quinto estado observado en una captura

Por último se muestra un diagrama con todos los estados y las posibles transiciones entre ellos con el objetivo de ofrecer una explicación clara de el algoritmo planteado:



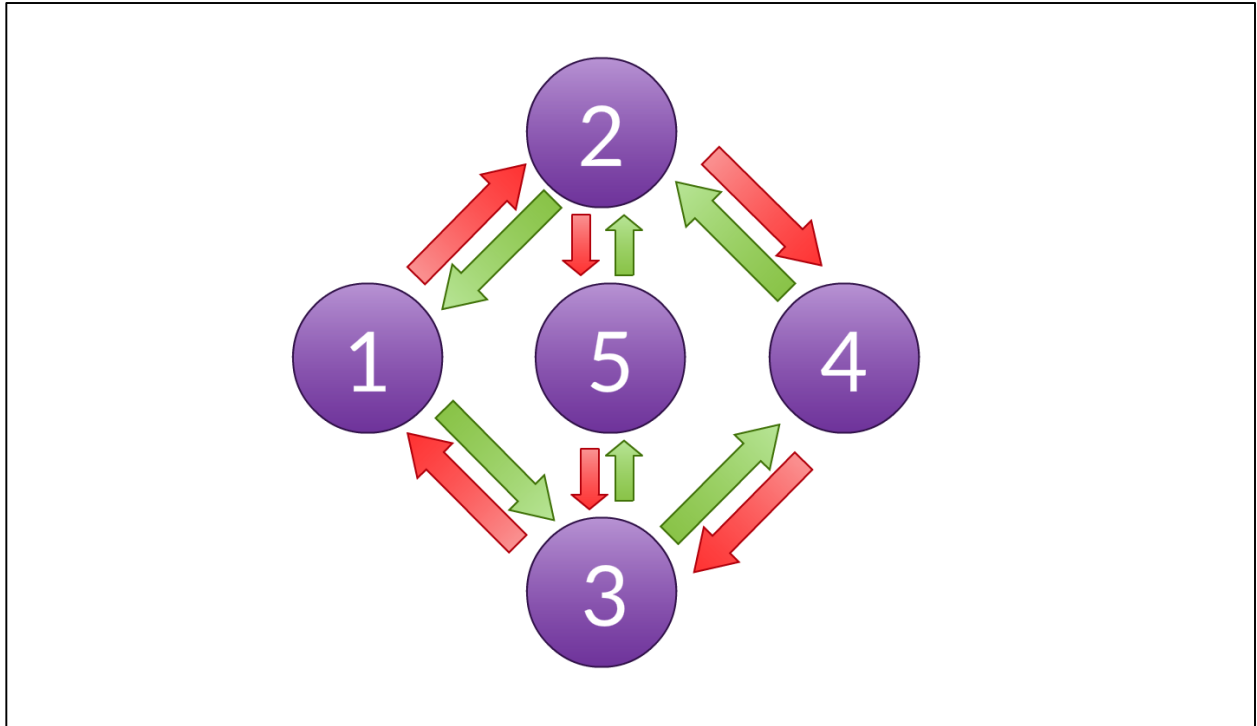


Figura 5-9. Relaciones entre los estados

A partir de estas transiciones se pudo generar un algoritmo para ir recorriendo los ficheros de texto plano con los paquetes y para poder ir identificando en que estado se encuentra la estación en cada momento. Una de las ventajas de la subdivisión del problema en estados es que conocemos muchos de los intervalos de actividad/ociosos (SIFS,ACK,RTS) que se repiten de forma fija en cada estado, por lo que conociendo el número de veces que se ha entrado en cada estado, se pueden obtener fácilmente los valores totales de el tiempo de transmisión, recepción, ocioso y sueño.

Esto se debe matizar, ya que realmente, muchas de las medidas que se toman en el programa no se basan en valores fijos, sino que se toman valores basados en la diferencia entre tiempos de los paquetes, como por ejemplo el tiempo entre que la estación se duerme y se despierta

Para obtener los valores se utilizan acumuladores que van acumulando las medidas en cada iteración del algoritmo y una serie de variables bandera que permiten identificar el estado en el que se encuentra la estación en cada momento de ejecución.

Las características principales de este algoritmo son las siguientes:

- Se emplean una serie de variables que permiten identificar tanto el estado en el que se encuentra la estación como la fase (esperando ACK, esperando RTS,etc) del estado en la que se encuentra. La variable *status* contiene el estado (1-6), y la variable *statXphase* contiene la fase en la que se encuentra para el estado X.
- Con cada paquete se comprueba el estado anterior (valores de *status* y *statXphase*), se acumulan los valores pertinentes y se actualizan el estado y/o la fase según corresponda.
- Cuando la estación envía un ACK al AP tras haber recibido un mensaje del AP con EOSP=1 (se acaba el periodo de servicio), se asume que la estación entra en estado de sueño hasta que se despierte, por alguna de las causas previamente planteadas.
- Una vez finalizada la ejecución dispondremos de el número de veces que la estación pasó por cada estado.

El cálculo del delay de cada paquete se calcula con la siguiente expresión:

$$(\text{Time} - \text{Firsttrptime}) - \left( \left( \frac{\text{Timestamp} - \text{Firsttrptimestamp}}{\text{samplingfreq}} \right) - \text{firstdelay} \right)$$

donde Time es el tiempo de llegada del paquete, Firsttime el tiempo de llegada del primer paquete rtp, timestamp es la marca de tiempo del paquete rtp recibido, firsttimestamp la marca de tiempo del primer paquete rtp recibido y samplingfreq la frecuencia de muestreo. Firstdelay representa el delay del primer paquete rtp recibido, obtenido como el valor medio del delay medido en la red. Debe tenerse en cuenta que este cálculo se realiza por separado para los paquetes ascendentes y descendentes y al finalizar se promedia el delay de todos los paquetes de la sesión.

El cálculo del jitter se deriva del cálculo del delay y se obtiene con la siguiente expresión [4]:

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16$$

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

donde  $S_i$  es la marca temporal del paquete  $i$ -ésimo,  $R_i$  el tiempo de llegada del paquete  $i$ -ésimo,  $D(i,j)$  el delay entre llegadas de el paquete  $i$ -ésimo y  $j$ -ésimo y  $J(i)$  el jitter medido para el paquete  $i$ -ésimo. Al igual que en el caso anterior, se calcula por separado para los paquetes ascendentes y descendentes y al finalizar se promedia el jitter de todos los paquetes de la sesión. [21]

Para el cálculo del consumo energético en las capturas en las que no se utilizó UAPSD se planteó otro algoritmo más sencillo, que se integró en UAPSD.c igualmente. Al igual que en el caso anterior, se diseñó tras analizar el intercambio de mensajes en los ficheros de captura de Wireshark.

Cabe recordar que, como se comentó anteriormente, al desactivar UAPSD en el interfaz inalámbrico, este no utilizaba el modo de ahorro PSM en el caso de transmitir y recibir tráfico en tiempo real como las llamadas de VoIP. Partiendo de esta idea, se plantearon únicamente tres posibles estados para la estación: transmisión, recepción y ocioso.

En la figura mostrada a continuación, se indican las posibles transiciones entre dichos estados:

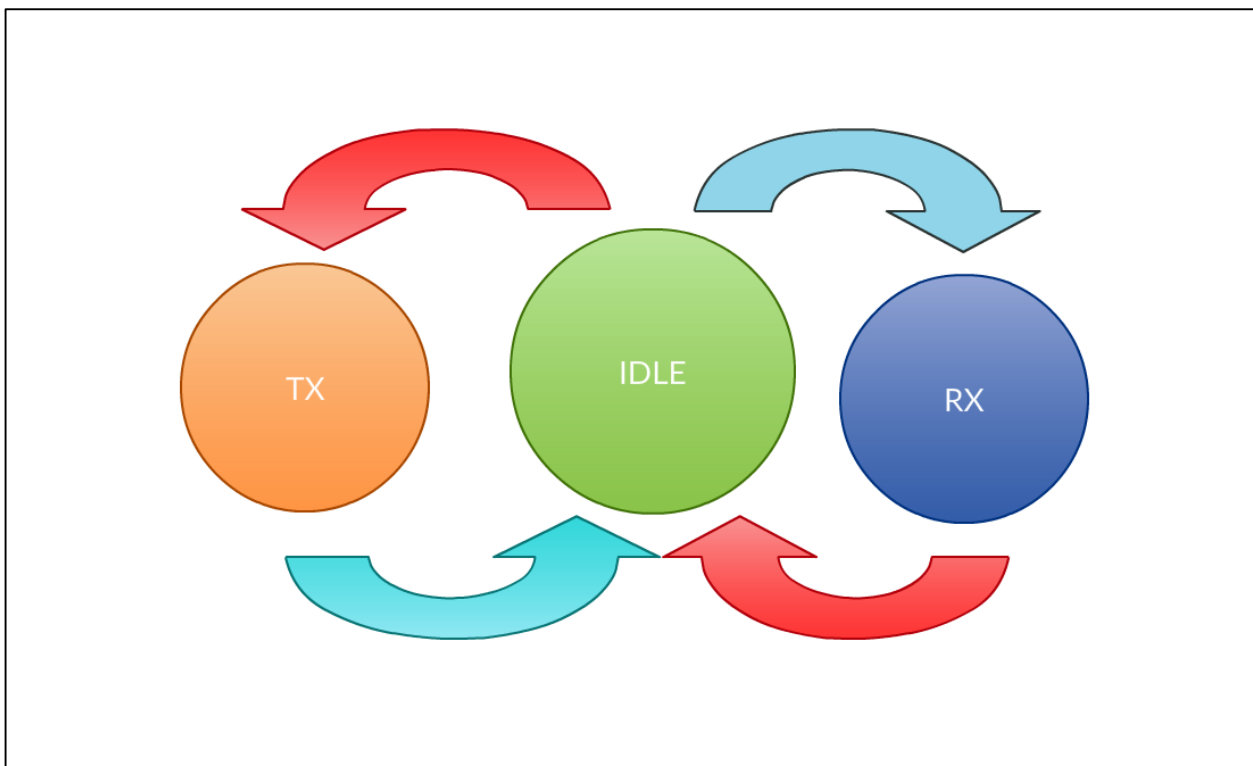


Figura 5-10. Relaciones entre los estados (sin UAPSD)

A partir de esta idea se diseñó el algoritmo a implementar, que computa el tiempo que a estación permanece en cada estado:

- Si el paquete es ascendente se computa y almacena el tiempo de transmisión (se conoce el régimen

binario del adaptador y la longitud de las tramas).

- Si el paquete es descendente se computa y almacena el tiempo de recepción (se conoce el régimen binario del adaptador y la longitud de las tramas).
- En el caso de las tramas Beacon del AP, se computa y almacena el tiempo de recepción, ya que al no activar UAPSD, la estación escucha periódicamente estas tramas.
- En los intervalos entre transmisión y recepción se asume que el interfaz radio está ocioso. Para calcular dichos intervalos se almacena el tiempo en el que se transmitió o recibió el último paquete, y se obtiene el intervalo como:

$$I_i = T_A - T_U - T_{TX/RX}$$

donde  $I_i$  es el intervalo de tiempo en el que la estación estuvo en estado ocioso,  $T_U$  el tiempo en el que se transmitió o recibió el último paquete y  $T_A$  el tiempo en el que se ha recibido o transmitido el paquete actual, finalizando el estado ocioso en el que previamente se encontraba.  $T_{TX/RX}$  es el tiempo durante el que se ha transmitido o recibido el paquete. Se utiliza este ajuste porque de lo contrario el cómputo del tiempo total en el estado ocioso sería la duración de la llamada completa.

Al igual que en el caso anterior, el programa computa el tiempo total de llamada, con el objetivo de corroborar la validez de los resultados.

Para el cálculo del jitter y el delay se sigue exactamente el mismo procedimiento que en el caso anterior, promediando los valores al finalizar la ejecución.

Para obtener los valores se utilizan acumuladores que van acumulando las medidas en cada iteración del algoritmo y una serie de variables bandera que permiten identificar el estado en el que se encuentra la estación en cada momento de ejecución.

### 5.1.1.2 Desarrollo y depuración

El desarrollo de la aplicación fue incremental y modular. Las diferentes funcionalidades y cálculos se fueron añadiendo por separado, y corroborando si efectivamente los datos obtenidos eran fiables. Por ejemplo, para el recuento total de paquetes se utilizaron las estadísticas de Wireshark, comprobando que el número de paquetes RTP de cada sesión de voz coincidía con el devuelto por el programa.

Se utilizaron así mismo acumuladores “temporales” que permitiesen ir comprobando si los valores eran correctos, así como los valores máximos y mínimos de los parámetros, y el número de paquete en el que se encontraban dichos valores, con el objetivo de poder depurarlo en el caso de encontrar valores no adecuados.

El primer códec para el que se desarrollaron todas las funcionalidades de forma completa fue PCMA, y posteriormente se adaptaron para iLBC y Speex, teniendo en cuenta que el formato de paquete cambia, y que el caso de Speex la cabecera de radio es diferente (radiotap header).

La parte más compleja del proceso de depuración fue la de corroborar que las transiciones entre estados se realizaban correctamente, ya que al utilizar muchas variables de bandera, cualquier error en una asignación de valor podía generar que algoritmo se quedase bloqueado en un estado sin pasar al siguiente.

Así mismo se encontraron varios problemas para el cálculo del delay, ya que los valores obtenidos eran negativos. Posteriormente, y tras realizar un análisis en profundidad de la situación se pudo concluir que no se estaba teniendo en cuenta el delay del primer paquete rtp cuyo timestamp se tomaba como valor inicial.

### 5.1.1.3 Parámetros y resultados

El programa que se desarrolló, de nombre UAPSD, cuyo código fuente se encuentra en el CD adjunto en la ruta */Software/UAPSDC.c*, se invoca con la siguiente sintaxis:

*UAPSD Fichero Nfpp Códec Modo*

- Fichero es el nombre del fichero de texto plano en el que se ha volcado el fichero de capturas de Wireshark.
- Nfpp indica en número de tramas por paquete empleado, que puede ir de 1 a 10, salvo en el caso del

códec iLBC, que sólo llega hasta 7

- Códec indica el códec empleado (pcma,ilbc,speex)
- Módo indica si se emplea el modo de ahorro de energía (uapsd) o no (nouapsd)

Respecto a los resultados generados por el programa, hay que señalar que tras la ejecución del mismo se muestra un resumen con las siguientes estadísticas en ambos sentidos de la comunicación (cuando corresponda), es decir para paquetes ascendentes y descendentes:

- Número de retransmisiones
- Número de paquetes RTP
- Porcentaje de pérdida de paquetes
- Tamaño máximo y medio de la ráfaga de pérdidas
- Duración máxima y media de la ráfaga de pérdidas
- Tiempo total de transmisión
- Tiempo total de recepción
- Tiempo total en ocioso (idle)
- Duración media de los paquetes RTP
- Delay máximo y medio
- Jitter máximo y medio
- Número de veces en cada estado
- Tiempo total de sueño
- Duración máxima y media del periodo de sueño
- Periodos totales de sueño
- Duración media del tiempo en despertar (QoS Null)

## 5.1.2 Medidas de QoS

Para la evaluación de la calidad de servicio obtenida se utilizaron tres métodos diferentes: dos programas para la evaluación perceptual de la calidad del habla (PESQ) basados en las normas P.563 [33] y P.862 [34] de la ITU, y el modelo-E.

### 5.1.2.1 PESQ

La evaluación perceptual de la calidad del habla es una familia de estándares que proporcionan una metodología para la evaluación automática de la calidad de la llamada tal y como la percibiría un usuario del sistema de telefonía.

PESQ emplea muestras reales de voz como señales para la evaluación de la calidad. Con el objetivo de caracterizar la calidad tal y como la percibiría un usuario humano, es de gran importancia probar los sistemas de telecomunicación con señales de voz reales.

Dependiendo de la información que emplea el algoritmo, los algoritmos de testeo de calidad de voz pueden clasificarse en dos categorías principalmente [22]:

- Los algoritmos “full reference” (FR) que utilizan una señal de referencia para realizar una comparación con la señal obtenida tras atravesar el sistema de telefonía. Las medidas que ofrecen son las más precisas, pero solo pueden ser aplicadas para tests dedicados en redes reales.
- Los algoritmos “no reference” (NR) que solo utilizan la señal degradada para la estimación de la calidad de servicio sin tener una referencia de la señal original. Estos algoritmos proporcionan estimaciones de

poca precisión ya que las características originales de la voz, como el ruido de fondo o la ausencia de voz en la fuente son completamente desconocidas.

Los resultados que PESQ proporciona se miden en la escala MOS (mean opinion score). Esta que mide la calidad experimentada de la llamada escala va desde 1(mala calidad) hasta 5(calidad excelente).

| MOS | Quality   | Impairment                   |
|-----|-----------|------------------------------|
| 5   | Excellent | Imperceptible                |
| 4   | Good      | Perceptible but not annoying |
| 3   | Fair      | Slightly Annoying            |
| 2   | Poor      | Annoying                     |
| 1   | Bad       | Very Annoying                |

Figura 5-11. Escala MOS [22]

#### 5.1.2.1.1 P.563

Se trata de un método de un solo extremo para la evaluación objetiva de la calidad vocal en aplicaciones de telefonía, por lo que emplea algoritmos NR.

La implementación que se empleó para evaluar la calidad de servicio de las llamadas es la desarrollada por la propia ITU. Se trata de un programa escrito en C que puede encontrarse en el CD adjunto en la ruta */Software/T-REC-P.563-200405-I!!SOFT-ZST-E*. Para su instalación puede consultarse el Anexo B. Este software incluye un conjunto de grabaciones de prueba y un pdf con el desarrollo de la norma que se consultó previamente para la preparación de las muestras de voz a utilizar para la evaluación.

Los requisitos principales que deben cumplir las muestras de voz son los siguientes [23]:

- Frecuencia de muestreo de 8KHz (en el caso de emplear frecuencias mayores es necesario utilizar un filtro paso bajo previamente).
- Debe haber un mínimo periodo de actividad vocal en la grabación de 3s.
- La grabación no debe superar los 20 segundos.
- El ratio de habla en la grabación debe superar el 25%.
- El ratio de habla en la grabación debe ser inferior al 75%.

Para cumplir con estos requisitos se tomaron dos muestras de cada grabación. Para las grabaciones almacenadas en el PC1 desde el minuto 0:26.5 hasta el 0:42.5 y desde el minuto 9:03 hasta el 9:19. Para las grabaciones recogidas en el PC2 desde el minuto 0:16 hasta el 0:28 y desde el minuto 3:10 hasta el 3:29. Previamente se analizó el audio para comprobar que el ratio de habla estaba dentro de los límites establecidos.

Una vez que se obtienen las muestras para su análisis, se ejecuta el programa con el siguiente comando:

```
p563 audiofile.wav -out outfile
```

donde *audiofile.wav* es el fichero de audio preparado para su evaluación y *outfile* el fichero en el que se desean volcar los resultados del análisis.

Con el objetivo de automatizar las pruebas se codificó un script de bash para realizar la estimación de la calidad de servicio de todas las conversaciones en ambos extremos. Puede encontrarse en el CD adjunto en la ruta */Software/MOS\_Processing.sh*.

Este script realiza en primer lugar un recorte de las muestras que se desean emplear utilizando el software Sox que puede encontrarse en el CD adjunto en la ruta */Software/sox-14.4.2* y posteriormente ejecuta el análisis de las muestras con p563.

Tras la ejecución, los resultados pueden consultarse en el fichero *out\_p563.csv*, cuya ruta se puede variar modificando la variable *CSVFILE*. Las salidas se generan en formato *.csv* para poder exportarlas con facilidad a Excel y generar las gráficas posteriormente.

El script puede adaptarse con facilidad a cualquier equipo modificando las variables *RUTA* y *EXEC* que almacenan la ruta y el nombre del ejecutable respectivamente. Es importante señalar que el script espera encontrar las grabaciones organizadas de la siguiente manera:

*Carpeta\_contenedora/pcx/CODEC\_xxUAPSD/OUTPUTX.WAV*

donde *pcx* indica el pc en el que se recogió la grabación, *CODEC* el códec que se empleó, *XX* puede estar vacío (se emplea *UAPSD*) o ser *NO* (no se emplea *UAPSD*), y *OUTPUTX.WAV* es el fichero de audio grabado en el extremo donde *X* indica el valor de *Nfpp*.

#### 5.1.2.1.2 P.862

Se trata de un método de dos extremos para la evaluación objetiva de la calidad vocal en aplicaciones de telefonía, por lo que emplea algoritmos FR.

La implementación que se empleó para evaluar la calidad de servicio de las llamadas es la desarrollada por la propia ITU. Se trata de un programa escrito en C que puede encontrarse en el CD adjunto en la ruta */Software/T-REC-P.862-200511-1!Amd2!SOFT-ZST-E*. Para su instalación puede consultarse el Anexo B. Este software incluye un conjunto de grabaciones de prueba y un pdf con el desarrollo de la norma que se consultó previamente para la preparación de las muestras de voz a utilizar para la evaluación.

Los requisitos principales que deben cumplir las muestras de voz son los siguientes [23]:

- Duración máxima del fichero de audio de 20 segundos.
- Duración mínima (recomendada) del fichero de audio de 8 segundos.
- El ratio de habla en la grabación debe superar el 40%.
- El ratio de habla en la grabación debe ser inferior al 80%.

Para cumplir con estos requisitos se tomaron dos muestras de cada grabación y sus respectivas referencias de audio. Para las grabaciones almacenadas en el PC1 desde el minuto 1:32 hasta el 1:36 y desde el minuto 0:26 hasta el 0:42. Para las grabaciones recogidas en el PC2 desde el minuto 0:16 hasta el 0:19 y desde el minuto 3:10 hasta el 3:03. Previamente se analizó el audio para comprobar que el ratio de habla estaba dentro de los límites establecidos.

Una vez que se obtienen las muestras para su análisis, se ejecuta el programa con el siguiente comando:

*PESQ +8000 reference.wav degraded.wav*

donde *reference.wav* es el fichero de audio de referencia, *degraded.wav* el fichero de audio grabado en uno de los extremos y *+8000* indica que la frecuencia de muestreo es de 8KHz (también puede funcionar con frecuencia de muestreo de 16KHz).

Con el objetivo de automatizar las pruebas y al igual que en el caso anterior, se codificó un script de bash para realizar la estimación de la calidad de servicio de todas las conversaciones en ambos extremos. Puede encontrarse en el CD adjunto en la ruta */Software/MOS\_Processing\_2.sh*.

Este script realiza en primer lugar un recorte de los audios de referencia y posteriormente de las muestras que se desean emplear utilizando el software Sox que puede encontrarse en el CD adjunto en la ruta */Software/sox-14.4.2* y posteriormente ejecuta el análisis de las muestras con p563.

Tras la ejecución, los resultados pueden consultarse en el fichero *out\_p862.csv*, cuya ruta se puede variar modificando la variable *CSVFILE*. Las salidas se generan en formato *.csv* para poder exportarlas con facilidad a Excel y generar las gráficas posteriormente.

El script contiene las mismas variables para modificar las rutas que el anterior y espera encontrar la misma

estructura de carpetas para los audios.

No obstante es importante señalar que en con P.862 el audio de referencia debe ser un poco más corto que el audio degradado, ya que debido al retardo que introduce un cierto offset en la grabación, es posible que las degradadas empiecen más tarde que las de referencia. El programa realiza una alineación previa de los ficheros de audio, por ello, al utilizar un fichero degradado de una duración ligeramente superior, se asegura que el funcionamiento es el correcto. [23]

### 5.1.2.2 Modelo-E

Para evaluar la calidad del audio, la ITU-T provee un modelo para su cálculo mediante la recomendación G.107 [26], comúnmente conocida como modelo-E. La implementación de este modelo se realiza mediante la recolección de parámetros de retardo total (jitter y retardos de red) y pérdidas.

La ecuación principal del modelo determina el factor R, que es la puntuación de calidad resultante de la aplicación del modelo a los datos de entrada. Posteriormente, dicha puntuación puede ser traducible a una escala MOS. El factor R está constituido por  $R_0$ , que representa la señal a ruido básica que incluye fuentes de ruido tales como ruido de circuito y ruido de ambiente;  $I_d$  representa las degradaciones producidas por el retardo, y el factor de degradación efectiva del equipo  $I_{e-eff}$  incluye la degradación debida a pérdidas de paquetes.

$$R=R_0-I_d-I_{e-eff}$$

El término  $R_0$  se considera constante, su valor se ha fijado a 93,2. Esto implica que, aunque las degradaciones fuesen nulas, el máximo valor alcanzable por el factor es igual a dicha constante. En cambio, los valores de  $I_d$  e  $I_{e-eff}$  se subdividen en parámetros más específicos.

El factor de degradación por retardo posee una ecuación simplificada que se basa en evaluar el retardo de red que sufren los paquetes transmitidos, expresado en milisegundos. Este factor  $I_d$  se calcula entonces como

$$I_d=(0,024*d)+0,11*(d-177,3)*H(d-177,3)$$

donde  $d$  es el retardo total observado por los paquetes de VoIP, mientras que  $H$  es la función de Heavyside tal que  $H(x) = 0$  para  $x < 0$  y  $1$  para  $x \geq 0$ .

Una vez calculado el factor de degradación por retardo pasamos a hacer lo propio con el factor de degradación efectiva. Este depende de unos valores tabulados mediante pruebas para intentar aproximarse al valor real de la forma más precisa posible. Entre los valores tabulados se hallan el factor de degradación del equipo para pérdida de paquetes nula  $I_e$  y el factor de robustez contra pérdida de paquetes  $B_{pl}$ . Así, eligiendo los valores predefinidos para cada códec, y recogidos en la tabla 5-1, se obtiene  $I_{e-eff}$  como sigue

$$I_{e-eff} = I_e + (95 - I_e) * \frac{P_{pl}}{\frac{P_{pl}}{B_{urstR}} + B_{pl}}$$

donde  $P_{pl}$  es un parámetro variable adimensional que mide la probabilidad de pérdidas de paquetes (expresada en tanto por cien) que se produce en una conversación de VoIP. La variable  $B_{urstR}$  es un parámetro adimensional también que mide las pérdidas de paquetes producidas a ráfagas en una conversación. Los valores  $I_e$  y  $B_{pl}$  pueden obtenerse de la propia recomendación G.107 de la ITU-T [43] para los códecs G.711 e iLBC, mientras que para Speex fue necesario consultar una estimación para dichos parámetros derivada de un estudio realizado recientemente [44].

Tabla 5-1. Valores de  $I_e$  y  $B_{pl}$  para los distintos códecs

| Nombre | $I_e$ | $B_{pl}$ |
|--------|-------|----------|
| Speex  | 18    | 21       |

|                 |   |      |
|-----------------|---|------|
| G.711<br>(PCMA) | 0 | 25,1 |
| iLBC            | 4 | 32   |

BurstR es el Burst ratio y es la última modificación añadida por la ITU-T para optimizar los resultados en base al comportamiento presentado por las pérdidas de paquetes. Relaciona la longitud de las ráfagas de pérdida de paquetes con la longitud de las ráfagas ante pérdidas aleatorias. Es por ello que siempre su valor es superior a la unidad, salvo cuando las pérdidas presentan una distribución aleatoria que entonces el resultado es uno.

Para medirlo de forma eficaz se define el tamaño medio de las ráfagas de paquetes perdidos  $B$ , medido en términos de tramas perdidas. Así, finalmente el parámetro *BurstR* queda expresado de la siguiente forma:

$$\text{BurstR} = \max\left\{1, B \cdot 1 - \frac{Ppl}{100}\right\}$$

El segundo término se corresponde con la propia expresión proporcionada por la recomendación para distribuciones de pérdidas de paquetes en cadenas de Markov de dos estados. [2]

## 5.2 Estadísticas obtenidas

### 5.2.1 Modelo-E

Para códec se almacenaron los siguientes datos de la salidas generadas por UAPSD.c , tanto para el canal ascendente como para el descendente, con y sin UAPSD:

- Porcentaje de pérdida de paquetes.
- Tamaño medio de ráfaga de pérdidas.
- El retardo medio.

Todos ellos pueden encontrarse en la ruta */Stats/RFACTOR.xlsx* del CD adjunto, un fichero de Excel donde se calcula el valor del factor R y posteriormente se mapea a la escala MOS [44].

A partir de los valores obtenidos se generaron las siguientes gráficas:



## 5.2.1.1 G.711 (PCMA)

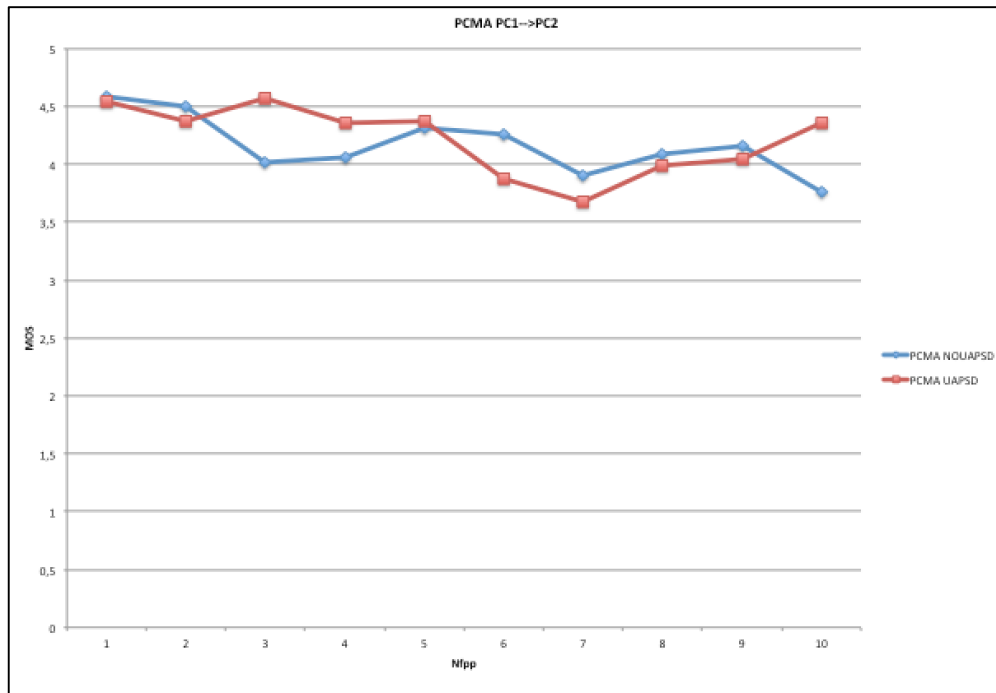


Figura 5-12. MOS obtenida frente a Nfpp del canal descendente.

En este primer caso puede apreciarse como el aumento de Nfpp influye sobre la calidad obtenida, tanto con UAPSD activado como sin él. Los puntos en los que el modo de ahorro de energía supera la calidad ofrecida por el modo normal pueden explicarse como resultado de un aumento del porcentaje de paquetes perdidos en las capturas para Nfpp=2,3,10 probablemente por la saturación del entorno inalámbrico durante un determinado intervalo de tiempo. No obstante, es necesario señalar que en el canal descendente se observa mayor variabilidad entre ambos modos para todos los códecs.

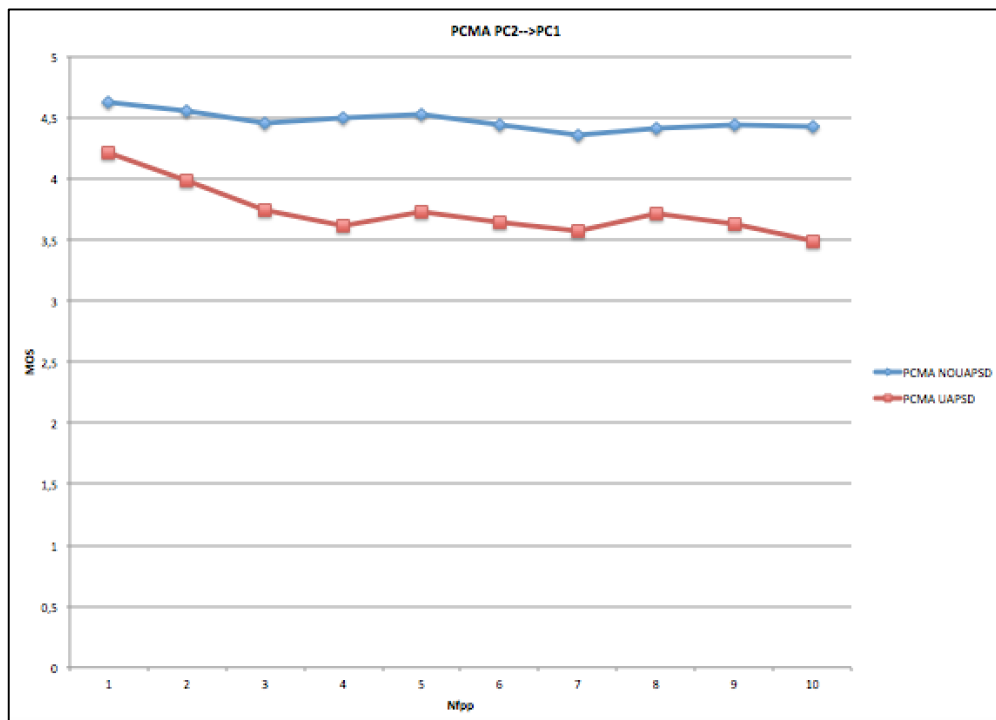


Figura 5-13. MOS obtenida frente a Nfpp del canal ascendente.

En el canal ascendente puede observarse nuevamente como el aumento de Nfpp disminuye el valor de MOS obtenido, en este caso hasta un mínimo de 3,49 para UAPSD y de 4,43 cuando no se utiliza este modo de

ahorro de energía. En este caso si se observa claramente como el empleo de UAPSD implica una menor calidad.

En resumen puede observarse que para cualquier caso con G.711 se obtiene una MOS superior a 3,5 (el peor de los casos, con UAPSD) y una MOS máxima de 4,53 (NOUAPSD).

### 5.2.1.2 iLBC

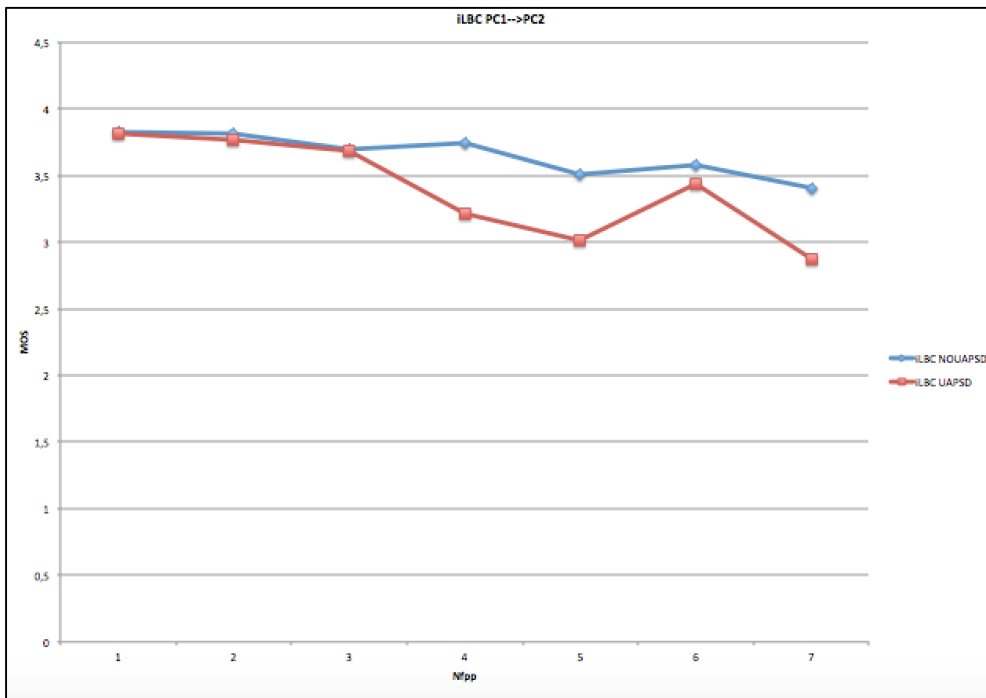


Figura 5-13. MOS obtenida frente a Nfpp del canal descendente.

Para el canal descendente en iLBC se vuelve a observar un comportamiento similar al del caso anterior, con un descenso de la calidad conforme se incrementa el número de tramas por paquete, que se acelera en el caso de utilizar UAPSD. No obstante, la diferencia de calidad para Nfpp=1,2,3 es mínima entre ambos modos de funcionamiento.

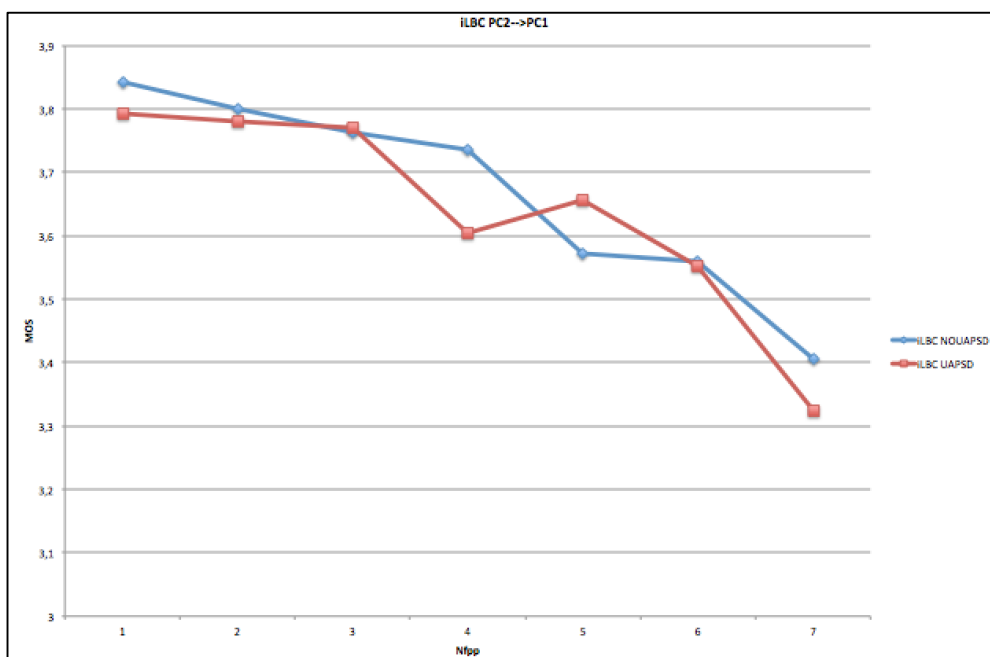


Figura 5-14. MOS obtenida frente a Nfpp del canal ascendente.

En el caso del canal ascendente, el comportamiento vuelve a ser similar al del códec anterior, aunque con UAPSD, este canal se ve menos afectado que al descendente. Los puntos en los que UAPSD mejora los valores que se obtienen sin UAPSD pueden explicarse así mismo como un aumento en la pérdida de paquetes para  $N_{fpp}=5$ , ya que el retardo medido en ambos modos es similar.

En resumen puede observarse que para cualquier caso con iLBC se obtiene una MOS superior a 2,8 (el peor de los casos con UAPSD), y una MOS máxima de 3,84 (NOUAPSD).

### 5.2.1.3 Speex

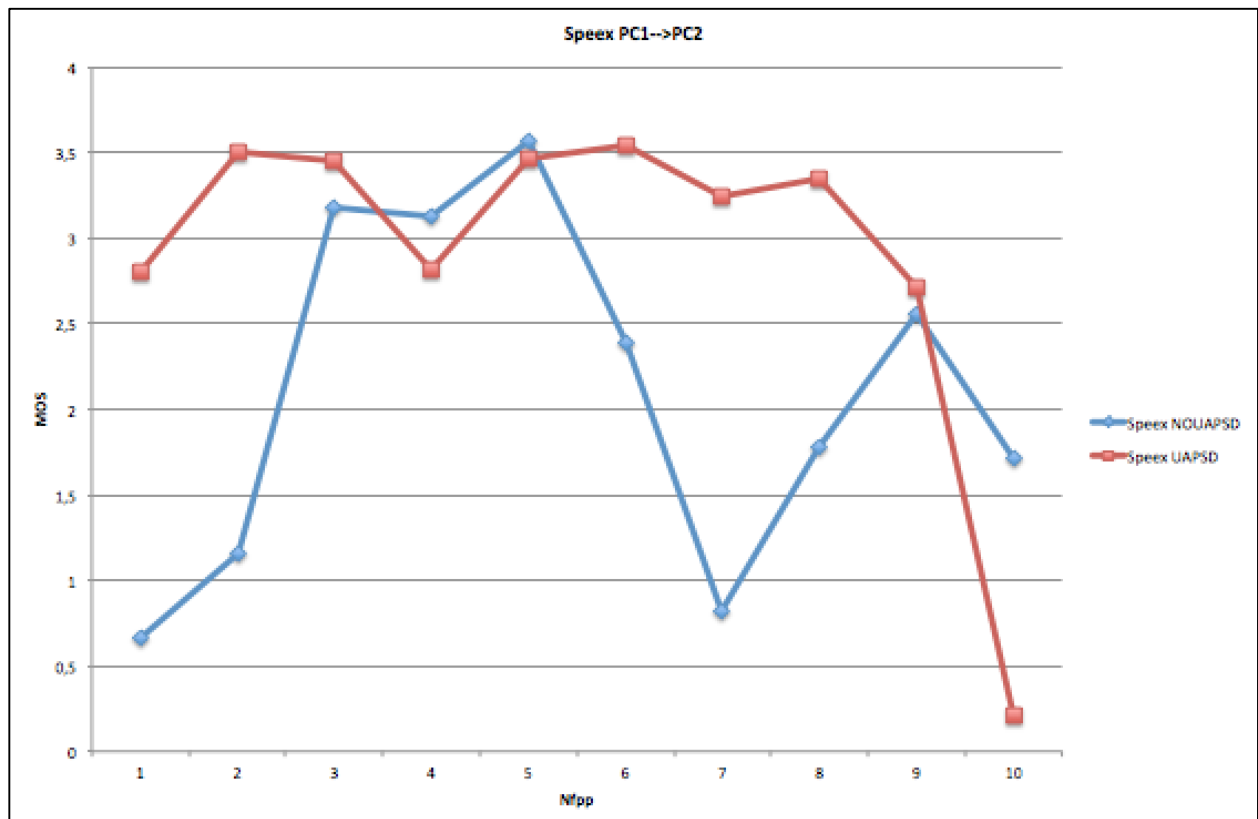


Figura 5-15. MOS obtenida frente a  $N_{fpp}$  del canal descendente.

En el caso del canal descendente para Speex, podemos observar un comportamiento anómalo, con varios picos y oscilaciones, esto se debe a una alta pérdida de paquetes (hasta un 18% para  $N_{fpp}=1$  y un 16% para  $N_{fpp}=7$ ) durante las capturas realizadas. El comportamiento más claro que podemos observar en este caso es el descenso en la calidad que se produce al incrementar el valor de  $N_{fpp}$ , sobre todo con UAPSD en funcionamiento.

La alta pérdida de paquetes en este caso puede tener su origen en situaciones de saturación del entorno inalámbrico durante la batería de pruebas, concretamente para  $N_{fpp}=1,2,7$ , e impide apreciar un comportamiento similar al de los casos anteriores.

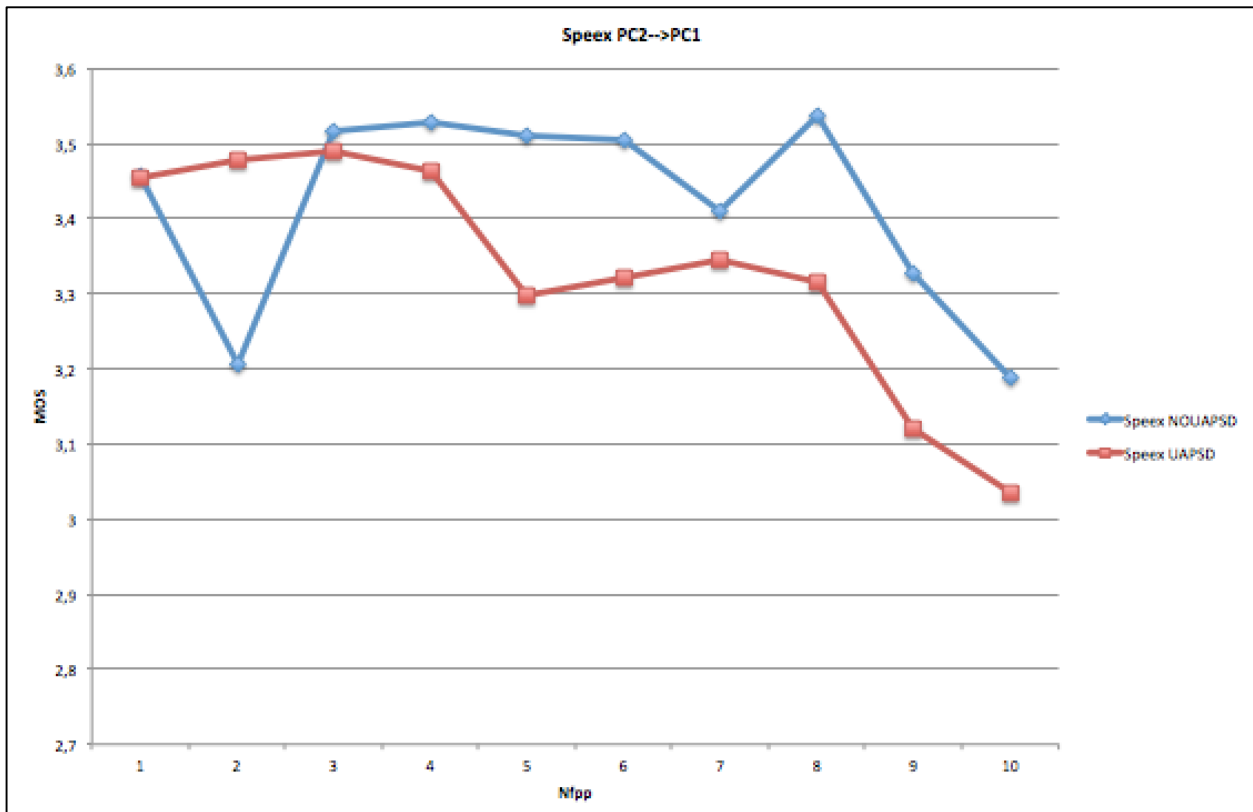


Figura 5-16. MOS obtenida frente a Nfpp del canal ascendente.

En esta gráfica puede observarse como se vuelve a repetir el descenso en la calidad conforme se aumenta el número de tramas por paquete empleado, que nuevamente se ve agudizado por el empleo de UAPSD.

Los valores mínimos y máximos para este códec son respectivamente 1 y 3,52, es decir, Speex ofrece una calidad inferior a la de G.711 y a la de iLBC, aunque con este último la diferencia es menor. Es importante señalar que el valor 1, que es el menor que puede tomar la MOS se alcanza en uno de los picos de las gráficas anteriores y como consecuencia de un aumento en la pérdida de paquetes.

### 5.2.2 P.563

Tras analizar los ficheros csv de salida generadas por el script *MOS\_processing.sh* y exportar sus datos a un documento de Excel se pudo observar que las gráficas obtenidas no reflejaban el comportamiento esperado y observado con el Modelo-E a pesar de haber seleccionado fragmentos de audio que cumplieran con las especificaciones requeridas.

Se realizaron pruebas tomando los mismos recortes del audio original, y en algunos casos se ofrecían valores aceptables mientras que en otros casos los valores eran completamente erróneos.

Debido a la falta de tiempo se decidió no indagar más a cerca de una posible solución para centrar todo el esfuerzo en los demás apartados, aunque se sospecha que puede guardar alguna relación con el nivel de volumen de las señales procesadas, lo cual se asumía que sería ajustado por el propio software de la ITU.

### 5.2.3 P.862

Al igual que en el caso anterior, las estadísticas obtenidas proporcionaban datos erróneos, incluso utilizando referencias sin degradación, motivo por el cual no se incluyen tampoco las estadísticas obtenidas.

No obstante, los scripts y los ficheros de salida se incluyen en el CD adjunto para un posible desarrollo futuro en la ruta */Software/*.

### 5.2.4 Energía y periodos de sueño

Para códec se almacenaron los siguientes datos de la salidas generadas por UAPSD.c , tanto para el canal ascendente como para el descendente, con y sin UAPSD:

- Tiempo total de transmisión
- Tiempo total de recepción.
- Tiempo total en idle.
- Tiempo total de sueño (si corresponde).
- Número de periodos de sueño.
- Duración media del periodo de sueño.

Todos ellos pueden encontrarse en la ruta */Stats/ENERGY.xlsx* del CD adjunto, un fichero de Excel donde se calcula el consumo energético de la estación en cada llamada, así como el ahorro de energía que implica el empleo de UAPSD.

El cómputo del consumo energético en aquellos casos en los que no se emplea UAPSD se realiza con la siguiente fórmula:

$$\text{Totalpwr} = \text{Tmtime} * \text{txpwr} + \text{Rmtime} * \text{rxpwr} + \text{Idletime} * \text{idlepwr}$$

donde Tmtime, Rmtime e Idletime son el tiempo total en transmisión, recepción e idle respectivamente, y los demás parámetros la potencia consumida por la estación en cada estado.

Para el cómputo del consumo energético en los casos en los que se emplea UAPSD basta con añadir a la expresión anterior un término que compute el consumo durante los periodos de sueño:

$$\text{Sleeptime} * \text{sleppwr}$$

Se han escogido los siguientes valores típicos de potencias para una interfaz inalámbrica (los mismos que se emplearon en el trabajo que se toma como punto de partida) [2]:

- txpwr=1650 mW
- rxpwr=950 mW
- idlepwr=800 mW
- sleppwr=40mW

A continuación se muestran para cada códec una serie de gráficas y estadísticas en las que se puede observar la relación entre el número de intervalos de sueño y la duración de los mismos, el tiempo total de sueño de la estación y el porcentaje de ahorro energético. Dicho ahorro energético puede obtenerse con la siguiente expresión:

$$100 * \left( 1 - \left( \frac{\text{Totalpwr\_UAPSD}}{\text{Totalpwr\_NOUAPSD}} \right) \right)$$

## 5.2.4.1 G.711 (PCMA)

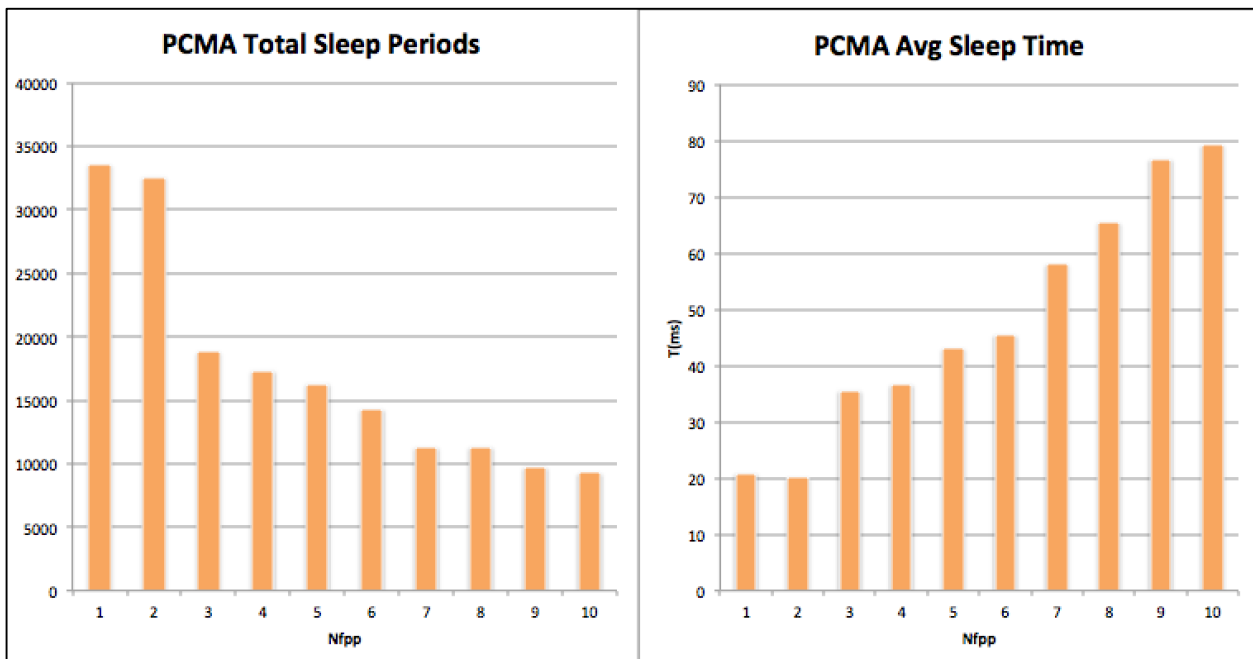


Figura 5-17. Periodos totales de sueño y duración media del periodo frente a Nfpp.

En esta primera gráfica puede observarse un comportamiento que se repetirá para todos los códecs; a medida que se incrementa Nfpp y con ello el retardo de empaquetado, desciende el número de intervalos totales de sueño mientras que aumenta la duración de los mismos.

El incremento del retardo de empaquetado se debe a que conforme crece Nfpp se deben muestrear más tramas antes de generar el paquete RTP, y concretamente, para G.711 con un periodo de muestreo de 10ms, y teniendo en cuenta los retardos que introduce la red, puede observarse como la estación aprovecha gran parte del tiempo para dormir (siempre que no disponga de paquetes para transmitir).

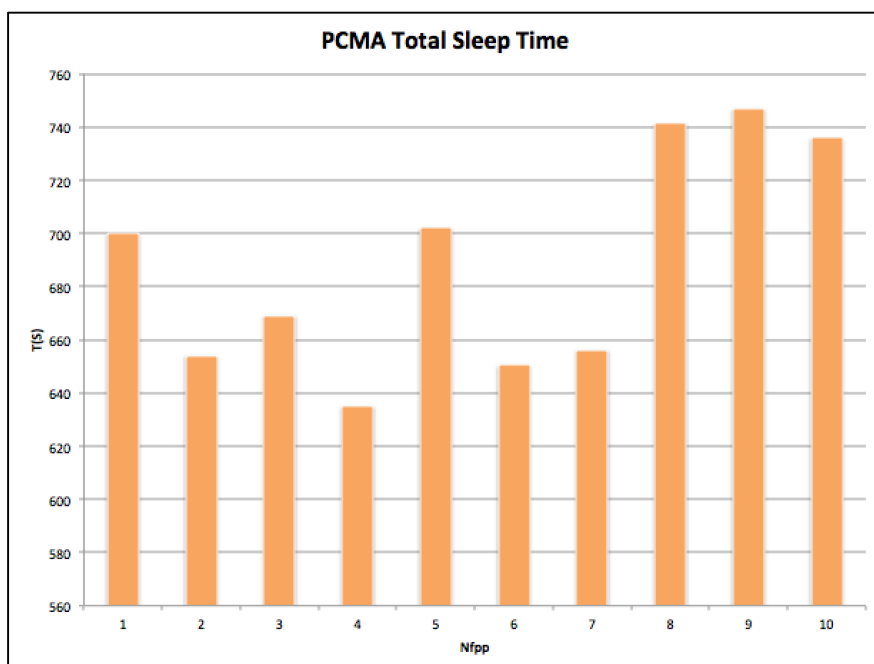


Figura 5-18. Tiempo total de sueño frente a Nfpp.

Respecto al tiempo total de sueño, el comportamiento será también similar en todos los códecs; conforme se incrementa Nfpp, tiende a subir, aunque en todos los casos pueden observarse ciertas oscilaciones o valores

inferiores, producto de las pérdidas que se producen durante algunas de las capturas, el tráfico adicional que manejan los interfaces inalámbricos (ARP,LLMNR) y patrones de tráfico variables.

Tabla 5-2. Porcentaje de ahorro de energía para G.711 con UAPSD

| Nfpp | Consumo<br>sin UAPSD<br>(mW) | Consumo<br>con UAPSD<br>(mW) | Ahorro de<br>Energía<br>(%) |
|------|------------------------------|------------------------------|-----------------------------|
| 1    | 731612,0401                  | 200326,138                   | 72,61852908                 |
| 2    | 726357,7739                  | 230899,0277                  | 68,21139168                 |
| 3    | 724971,2678                  | 217070,8961                  | 70,05800013                 |
| 4    | 724244,3396                  | 241590,4867                  | 66,64240595                 |
| 5    | 723531,6727                  | 190396,7355                  | 73,68508627                 |
| 6    | 723330,9901                  | 229370,227                   | 68,28972764                 |
| 7    | 722981,0216                  | 224409,5093                  | 68,96052558                 |
| 8    | 722753,9731                  | 159283,4953                  | 77,96158842                 |
| 9    | 722584,7195                  | 155197,3678                  | 78,52191396                 |
| 10   | 722467,1262                  | 163153,9077                  | 77,41711673                 |

#### 5.2.4.2 iLBC

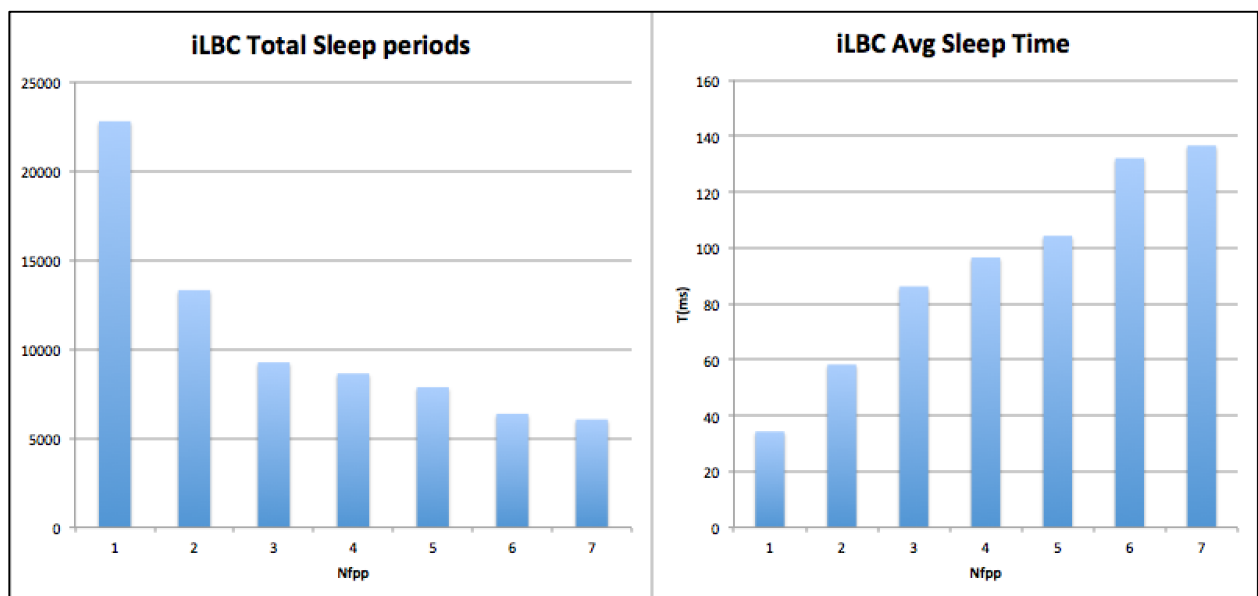


Figura 5-19. Periodos totales de sueño y duración media del periodo frente a Nfpp

En este caso observamos nuevamente el comportamiento previamente expuesto. iLBC emplea un periodo de muestreo de 30ms, por lo que nuevamente se puede observar que la estación aprovecha una parte importante del tiempo de muestreo para dormir.

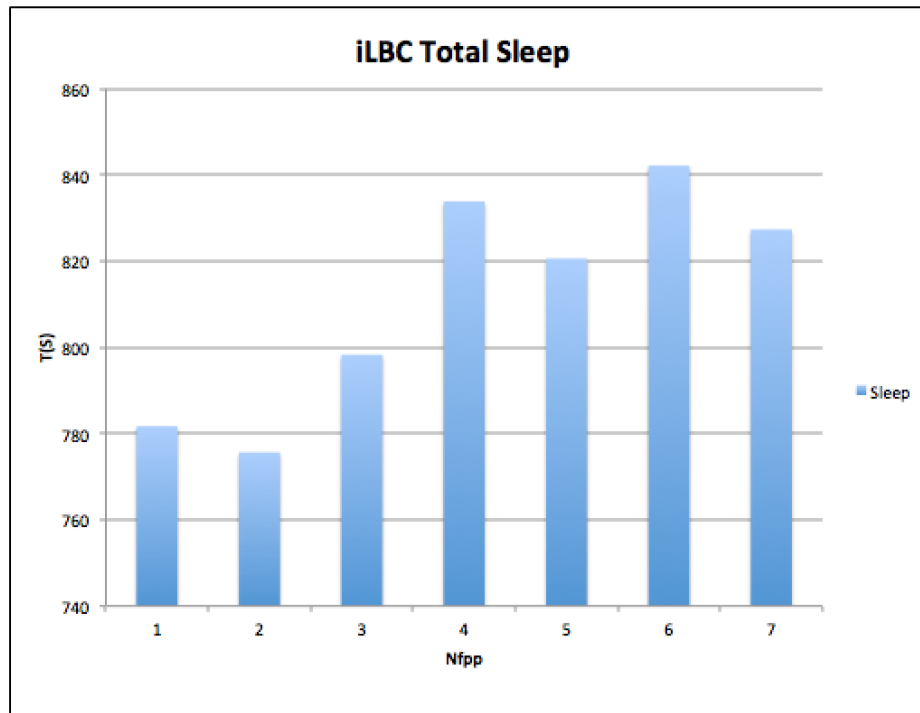


Figura 5-20. Tiempo total de sueño frente a Nfpp

El tiempo total de sueño muestra nuevamente un comportamiento similar, con una tendencia ascendente aunque se vuelven a observar ciertas oscilaciones.

Tabla 5-3. Porcentaje de ahorro de energía para iLBC con UAPSD

| Nfpp | Consumo sin UAPSD (mW) | Consumo con UAPSD (mW) | Ahorro de Energía (%) |
|------|------------------------|------------------------|-----------------------|
| 1    | 722.893                | 128802,6733            | 82,18233181           |
| 2    | 720714,5286            | 132078,1643            | 81,67399725           |
| 3    | 720946,8503            | 114695,0915            | 84,09104756           |
| 4    | 720488,0411            | 87129,71334            | 87,90684808           |
| 5    | 720389,0393            | 97151,63303            | 86,51400455           |
| 6    | 720292,6387            | 80567,33651            | 88,81463836           |
| 7    | 720326,5282            | 91808,69446            | 87,25457263           |

Respecto al ahorro energético, puede observarse una mejora con respecto al ofrecido por G.711, aunque es necesario tener en cuenta que iLBC sólo admite como máximo un valor de Nfpp=7.



5.2.4.3 Speex

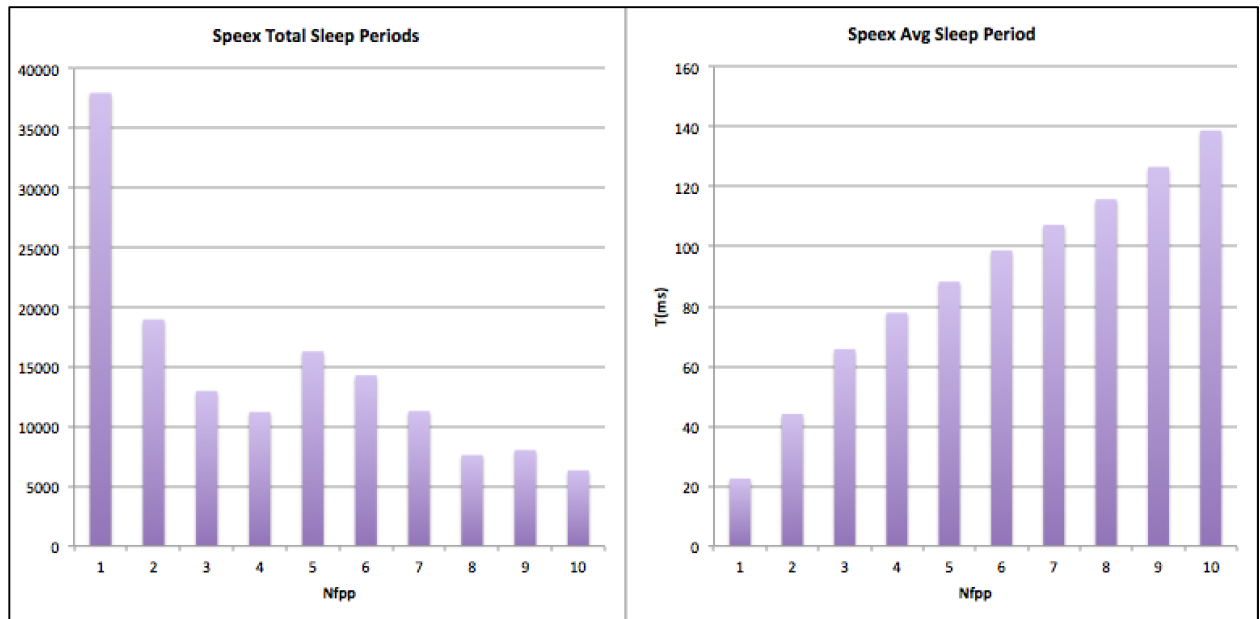


Figura 5-21. Periodos totales de sueño y duración media del periodo frente a Nfpp

Se puede observar una vez más una tendencia similar a la de los casos anteriores. Speex utiliza un periodo de muestreo de 20ms por lo que puede observarse como se aprovecha gran parte del tiempo para dormir.

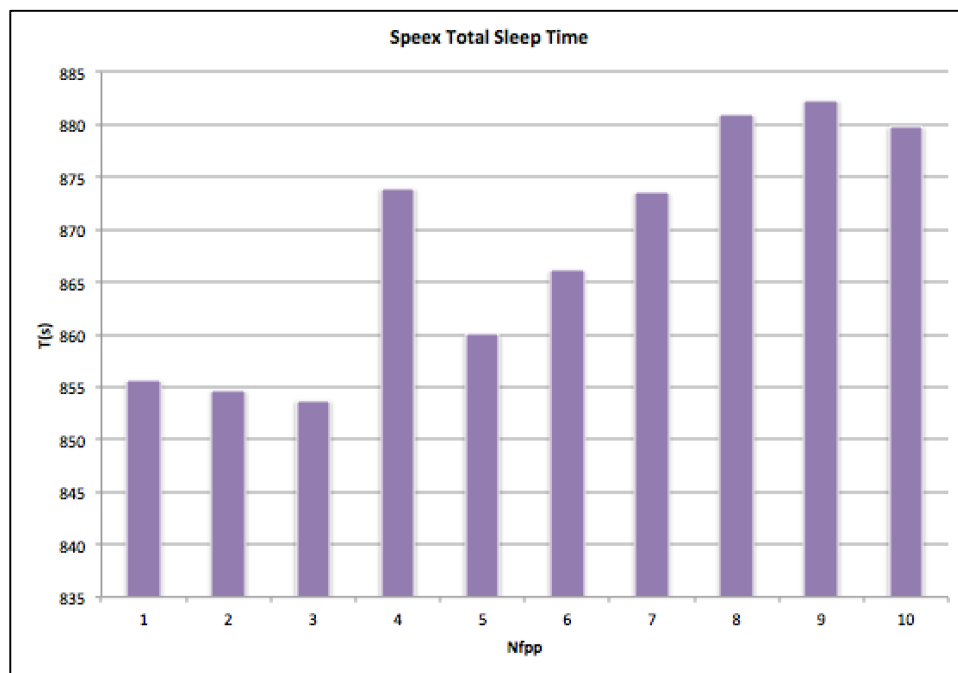


Figura 5-22. Tiempo total de sueño frente a Nfpp

En esta figura se puede volver a observar la tendencia ascendente, con tiempos totales de sueño superiores a los de los otros códecs previamente analizados.

Tabla 5-4. Porcentaje de ahorro de energía para Speex con UAPSD

| Nfpp | Consumo sin UAPSD (mW) | Consumo con UAPSD (mW) | Ahorro de Energía (%) |
|------|------------------------|------------------------|-----------------------|
| 1    |                        |                        |                       |
| 2    |                        |                        |                       |
| 3    |                        |                        |                       |
| 4    |                        |                        |                       |
| 5    |                        |                        |                       |
| 6    |                        |                        |                       |
| 7    |                        |                        |                       |
| 8    |                        |                        |                       |
| 9    |                        |                        |                       |
| 10   |                        |                        |                       |

|    |             |             |             |
|----|-------------|-------------|-------------|
| 1  | 723587,9105 | 74554,54023 | 89,69654701 |
| 2  | 721969,7472 | 72865,67423 | 89,90737846 |
| 3  | 721252,2009 | 72856,04691 | 89,8986725  |
| 4  | 720983,1468 | 57186,68407 | 92,06823567 |
| 5  | 720822,8099 | 67353,6755  | 90,65600109 |
| 6  | 720738,5646 | 62730,00002 | 91,29642798 |
| 7  | 720692,717  | 58698,92478 | 91,85520772 |
| 8  | 721425,3302 | 51227,12901 | 92,89917794 |
| 9  | 720538,9362 | 50126,93377 | 93,04313324 |
| 10 | 720507,6237 | 52004,1061  | 92,7822962  |

Respecto al ahorro, Speex es el códec que mayor porcentaje nos ofrece, aunque como se pudo observar previamente, la calidad ofrecida es menor a la de los anteriores.

Es importante comentar que en la capturas con UAPSD para Nfpp=5,7,8 se produjeron problemas durante las mismas, ya que la estación se desautenticó durante ciertos periodos y dejó de utilizar UAPSD, por lo que el cálculo total tuvo que hacerse en varios tramos. Por ello no se consideran estos valores totalmente fiables, y como veremos a continuación, no fueron empleados para determinar el mejor códec para una MOS dada.

### 5.2.5 Ahorro, calidad y códecs

Si bien al comienzo del proyecto se planteó la intención de la búsqueda de un punto de equilibrio entre el consumo energético y la calidad ofrecida, a la luz de los resultados anteriormente analizados, nos encontramos en disposición de realizar esta búsqueda.

Para ello se planteó definir una MOS mínima a garantizar, que se iría incrementando en intervalos de 0,2, en un rango de 2,5 a 4,5. Es importante señalar aquí que se ha considerado la MOS media obtenida entre el canal ascendente y descendente para cada caso.

A continuación se muestra una gráfica que ilustra los resultados de esta búsqueda, con el nombre del códec, el porcentaje de ahorro obtenido y el valor de Nfpp.

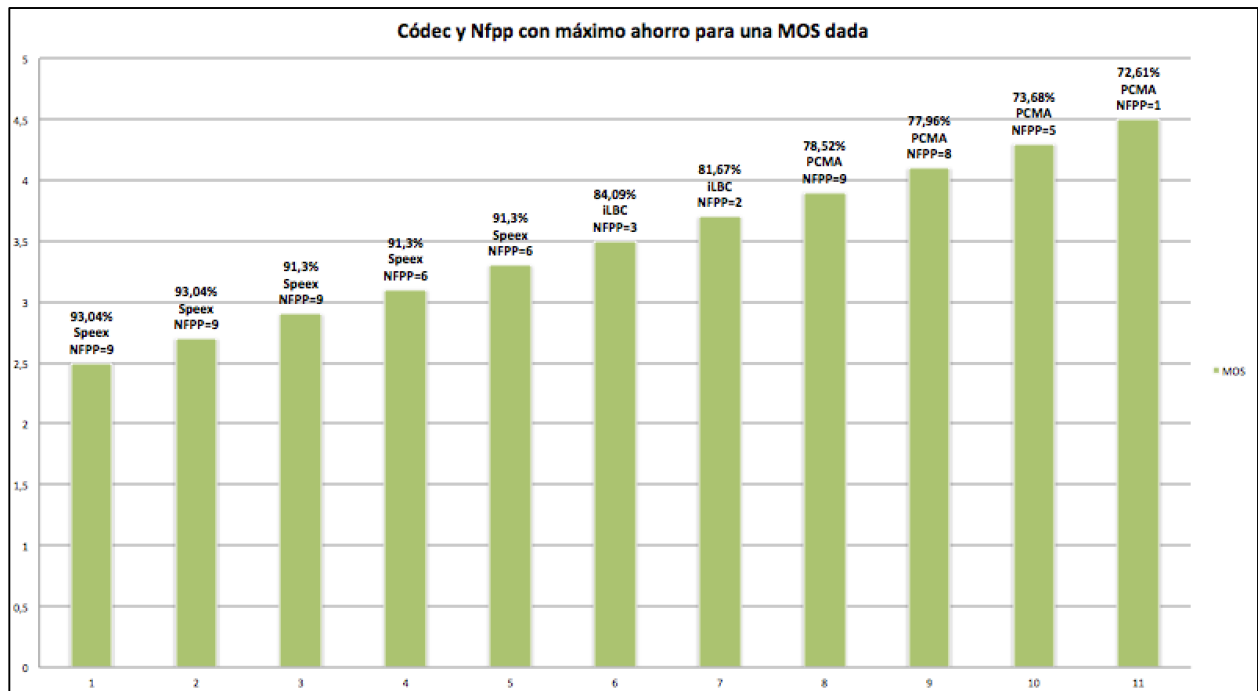


Figura 5-23. Mejor Códec,Nfpp en función del ahorro para una MOS mínima

Puede observarse que cuando los requisitos de calidad son bajos, el códec que mayor ahorro proporciona es Speex. De hecho, se mantiene como el mejor hasta llegar a 3,5, donde ya no es capaz de proporcionar suficiente calidad, y proporciona un ahorro energético de entre el 93,04% y el 91,3%.

Es importante señalar que como se comentó anteriormente, no se han tenido en cuenta los datos de consumo de Speex para Nfpp=7,8 debido a que se encontraron errores en las capturas, pero probablemente, de haber dispuesto de los mismos, la transición en el Nfpp no sería tan brusca, ya que se pasa de 9 a 6.

Cuando se exige una MOS superior a 3,3 el códec que proporciona un mayor ahorro energético garantizando el nivel de calidad es iLBC, con un valor de Nfpp=3. Con el siguiente incremento (3,7), el valor de Nfpp desciende en una unidad.

El ahorro energético con iLBC se sitúa entre el 84,09% y el 81,67%.

Por último puede observarse como a partir de 3,9 el único códec que puede garantizar el nivel exigido es PCMA, pero como es de esperar, el ahorro energético es menor; entre un 78,52% y un 72,61%.

Puede observarse nuevamente como al aumentar los requisitos de calidad, el número de tramas por paquetes va descendiendo.

Si bien se planteaba la búsqueda de un punto de equilibrio, se debe matizar que realmente dependerá del nivel de calidad que estemos dispuestos a perder por conseguir un mayor ahorro energético, o bien de la cantidad de energía que dejemos de ahorrar por garantizar una determinada calidad de servicio, pero los valores obtenidos pueden servir como referencia orientativa para determinar el códec a utilizar y el valor de Nfpp en función de la situación.

## 5.2.6 Conclusiones

Una vez realizada la exposición y el análisis de los resultados obtenidos tras ser procesados con el software desarrollado, llega el momento de recapitular y mostrar las conclusiones obtenidas.

Cabe señalar que este proyecto se planteó inicialmente como un procedimiento de evaluación del ahorro energético, sin embargo, la fase de investigación se alargó bastante más de lo esperado, principalmente porque la mayoría de los dispositivos estudiados no respondían como se esperaba; de hecho solo uno de los adaptadores que se adquirieron proporciona el funcionamiento deseado.

Haber dedicado tanto tiempo a encontrar el dispositivo adecuado para las pruebas y el estudio de la configuración, tanto del escenario como del dispositivo, han tenido como consecuencia una disminución del

alcance del proyecto, ya que inicialmente se pretendían utilizar varios terminales realizando llamadas de VoIP de forma simultánea.

No obstante, y teniendo en cuenta dichas limitaciones, se ha intentado automatizar al máximo todo el procedimiento de recogida de datos y su posterior procesamiento. También se ha llevado a cabo una labor de documentación exhaustiva de todos los pasos seguidos para que los experimentos se puedan repetir, con uno o varios terminales.

En cuanto a las conclusiones principales del proyecto, se pueden señalar varias:

Inicialmente se planteó la intención de comprobar la eficiencia de los diferentes modos de ahorro de energía, y con este propósito, se ha podido determinar que el empleo de UAPSD tiene como consecuencia una reducción importante del consumo energético de la estación. Se han obtenido valores que oscilan entre un 70% y un 90% de mejora con respecto a las configuraciones que no lo utilizan.

El porcentaje de ahorro entre el empleo de PSM o no, se ha podido obtener de los resultados del trabajo que se toma como punto de partida [2]. Los ficheros proporcionados para su cálculo son los mismos que se entregaron con el proyecto, y pueden encontrarse en el CD adjunto en la ruta /Stats/Calc.xlsx y /Stats/Calc-psm.xlsx. El cómputo se realizó para  $N_{fpp}=2,16$  y para un  $N_{fpp}$  variable, aunque realmente el único valor relevante es  $N_{fpp}=2$  ya que es el único que se puede comparar con los resultados de este proyecto.

Con dicho valor de  $N_{fpp}$ , utilizando G.711 se obtuvo un ahorro de 57,38%, mientras que con UAPSD este valor es de 68,21%, por lo que el ahorro con UAPSD es superior. Sin embargo, y como se comentó anteriormente, estos cálculos de energía consumida se obtuvieron de forma teórica, ya que como se ha podido comprobar de forma práctica, PSM no funciona con tráfico en tiempo real (por lo menos con la configuración y los dispositivos empleados).

No obstante, es un dato interesante y a tener en cuenta, ya que se ha podido comprobar que UAPSD es el modo de ahorro de energía idóneo para tráfico en tiempo real, y como se comentó durante el desarrollo teórico, el estándar 802.11e establece que cada categoría de acceso puede utilizar el modo de ahorro que desee. Así el empleo de ambos mecanismos, en función del tráfico que se vaya a cursar es una buena opción para maximizar el ahorro de energía.

A pesar de haber obtenido unos datos de ahorro de energía tan elevados es necesario tener en cuenta que el escenario empleado ha sido especialmente favorable, ya que únicamente una estación estaba conectada al AP, y el ancho de banda disponible era más que suficiente para cualquier códec, por lo que se puede entender que conforme aumenta el número de estaciones, el ahorro de energía disminuirá.

Tras analizar el comportamiento de la estación, se ha podido observar que durante una parte importante del periodo de muestreo, la estación duerme, por lo que a pesar de que varias estaciones empleen el mismo AP, el ahorro energético seguirá siendo importante, ya que en el momento que una consigue transmitir, duerme si no tiene nada más que enviar por lo que el número de estaciones contendientes será menor al número de estaciones totales.

Se ha podido observar también como el empleo de UAPSD tiene una repercusión directa en la calidad de la llamada, afectando más al canal descendente. Esto es así porque hasta que la estación no se despierta y solicita la información almacenada con una trama trigger (QoS Null o de datos), las tramas permanecen en el buffer de la categoría de voz, mientras que la estación puede despertarse en cualquier momento para enviar datos si dispone de ellos. Esto implica un retardo adicional para el canal descendente, lo que influye sobre el retardo total que percibe la aplicación.

No obstante el canal ascendente también se ve afectado por el incremento del valor de  $N_{fpp}$ , ya que como se puede observar; para valores grandes de  $N_{fpp}$ , la calidad desciende hasta alcanzar valores mínimos en la mayoría de los casos.

La duración de los intervalos de sueño y el número de intervalos está relacionado también con el valor de  $N_{fpp}$ , y como se comentó anteriormente, se debe principalmente a que la estación aprovecha los intervalos de muestreo (que crecen con  $N_{fpp}$ ) para dormir. Cuando mayor es el tamaño medio del periodo de sueño, menor es el número de periodos en los que la estación duerme y viceversa.

El tiempo total de sueño también tiene una cierta relación con respecto a  $N_{fpp}$ , ya que como se ha podido

observar, cada códec se mueve en un determinado rango de porcentajes que oscila en torno al mínimo entre un 3% y un 5% aproximadamente. En todos los códecs, a pesar de que se observan ciertas oscilaciones del valor, aparece una tendencia ascendente conforme se incrementa Nfpp.

A partir del acumulador del tiempo medio de sueño antes de despertar para enviar una trama QoS Null, se ha podido determinar así mismo que la estación despierta para escuchar los DTIM de la Beacon. Se ha podido observar que el valor máximo de sueño de la estación no suele superar (o si lo hace es en cantidades pequeñas) el intervalo de Beacon (500ms), que coincide así mismo con el intervalo de DTIM.

Se ha podido comprobar que el códec con mayor bitrate (G.711-64Kbps) es el que mayor consumo energético tiene aunque proporciona la mejor calidad. iLBC con un bitrate menor mejora los datos de consumo como es lógico y proporciona una menor calidad. Por último Speex, el único códec con bitrate variable es el que proporciona el mayor ahorro energético, aunque como se ha visto, a costa de una reducción mayor de la calidad.

A parte de las conclusiones principales, relacionadas con las estadísticas obtenidas, se ha podido comprobar que a pesar de que el empleo de UAPSD permite un ahorro de energía importante, la configuración previa de la red y los dispositivos es compleja, y se requieren dispositivos certificados con WMM-PS, aunque como se ha visto, en algunos casos ni aún así se garantiza el correcto funcionamiento.

Realmente la configuración de la infraestructura es la parte más sencilla, ya que los APs de altas prestaciones soportan WMM-PS, el problema aparece con los terminales de usuario, ya que el software que empleen debe estar preparado para realizar el marcado DSCP de los paquetes (en Windows se ha tenido que realizar a mano, pues pjsua de por sí no realizaba el marcado DSCP) y la interfaz inalámbrica debe ser también compatible con WMM-PS. Por tanto se plantea aquí una cuestión fundamental con respecto al equipamiento de usuario.

Analizando la tendencia actual, los smartphones se emplean en los puestos de trabajo para dar servicio de telefonía, y existen multitud de aplicaciones de VoIP para estos dispositivos. No obstante, tras la investigación inicial que se desarrolló para determinar los dispositivos a emplear en el proyecto, no se encontraron smartphones con soporte explícito para U-APSD, aunque sí teléfonos VoIP inalámbricos específicamente diseñados para ello.

Por tanto, y teniendo en cuenta la proliferación de los smartphones y el uso intensivo al que se someten (con el consecuente gasto de batería), la integración de WMM-PS/UAPSD en los mismos puede suponer un importante ahorro energético y el empleo de un único dispositivo para realizar llamadas, tanto por VoIP como por red móvil.

Respecto a los dispositivos probados, y como se ha comentado, cabe destacar que en la gran mayoría el modo de ahorro de energía no funciona, y en varios casos, a pesar de estar certificados con WMM-PS ni siquiera ofrecen opciones de configuración para activarlo. Puede deberse a que estén diseñados para interoperar con dispositivos del mismo fabricante, pero tras las pruebas realizadas se ha podido concluir que sólo uno de los ocho adaptadores es capaz de utilizar UAPSD. Cabe señalar también que aunque muchos de ellos son compatibles con Linux, muchas de las funcionalidades que ofrecen para Windows no pueden ser utilizadas.

### 5.2.7 Líneas de futuras de trabajo

Como se comentó anteriormente, debido a los problemas que fueron surgiendo, hay partes del proyecto que no se han podido llevar a cabo, por ello, se detallan a continuación una serie de puntos que se podrían ser desarrollados:

- Sería importante repetir las pruebas con más códecs, especialmente los que se pueden integrar utilizando las IPP de Intel con pjsip (se detallan en el anexo A). Realmente no se pudieron integrar porque no se consiguió la licencia necesaria, pero una vez que se consiga, la integración es bastante sencilla y viene detallada paso a paso en la web de pjsip.
- Otro punto importante a desarrollar sería el simular un escenario real con varios terminales inalámbricos. Las pruebas pueden repetirse y escalarse para más equipos con facilidad gracias a los scripts que se han desarrollado y a que se adquirieron 4 tarjetas inalámbricas de Intel. La complejidad reside en sincronizar los relojes de todos los equipos y el arranque de los scripts para que las llamadas se realicen en el instante adecuado, ya que con dos equipos era muy sencillo iniciar los dos scripts simultáneamente.

- Sería también importante identificar el origen de los problemas con los programas de evaluación perceptual basados en las normas P.563 y P.862, ya que gracias a los scripts desarrollados permiten generar estadísticas con una simple ejecución que apoyen las obtenidas a partir del modelo E.
- Como se comentó, no se pudo deshabilitar el empleo de RTS/CTS por parte del AP, un mecanismo empleado para tramas de gran tamaño, por lo que sería bueno repetir las pruebas con un AP en el que se pudiera deshabilitar para poder observar la repercusión sobre el canal descendente y sobre el intercambio de paquetes en general.
- Se ha podido observar que en la estación transmite como máximo 2 tramas seguidas durante TXOP debido a su duración, por lo que sería interesante probar diferentes valores de TXOP especialmente para valores bajos de Nfpp, que es donde se observa principalmente este patrón de envío de paquetes.
- El trabajo que se tomó como punto de partida [2] dio como resultado (entre otros) un software (modificación de pjsua) que se encarga de modificar el valor de Nfpp para alcanzar una determinada MOS objetivo. La realización de pruebas con este software con el empleo de UAPSD puede ser también un punto interesante a desarrollar.
- Sería también interesante emplear UAPSD para varias categorías de acceso con diferente prioridad para evaluar el ahorro energético y la calidad en función de la categoría de la que se trate.
- Las pruebas realizadas con RTS/CTS en la estación generaron un audio con muy mala calidad y mucho eco. Investigar el motivo de dicho comportamiento también es un punto en el que no se pudo investigar más. En el CD adjunto puede encontrarse una muestra de dicho audio con G.711 para Nfpp=1,5,7

---

## REFERENCIAS

---

- [1] Data solutions group. (2012) ersdatasolutions. [Online]. <http://www.ersdatasolutions.com/partners/vowlan>
- [2] Daniel Jiménez Lara, *Estudio experimental de calidad y consumo energético en VoIP sobre redes 802.11*. Sevilla, Sevilla, España: Universidad de Sevilla, 2013.
- [3] IETF. (1980, Aug.) ietf.org. [Online]. <https://www.ietf.org/rfc/rfc768.txt>
- [4] IETF. (2003, July) [Online]. <https://tools.ietf.org/html/rfc3550>
- [5] Catharine Adams. (2010, May) Slideshare. [Online]. <http://www.slideshare.net/Catharine24/voip-voice-over-ip-introduction-course>
- [6] IETF. (1981, Sep.) ietf.org. [Online]. <https://tools.ietf.org/html/rfc793>
- [8] IETF. (1998, June) [Online]. <http://tools.ietf.org/html/rfc2616>
- [7] IETF. (2002, June) [Online]. <https://www.ietf.org/rfc/rfc3261.txt>
- [9] IETF. (1987) [Online]. <https://www.ietf.org/rfc/rfc1035.txt>
- [10] David Carrero. (2006, Jan.) Messenger. [Online]. [http://messenger.es/sip\\_445/](http://messenger.es/sip_445/)
- [11] Changhua He and John C Mitchell. (2008) Security Analysis and Improvements for IEEE 802.11i. Pdf.
- [12] (2011, Mar.) revolutionwifi.net. [Online]. <http://www.revolutionwifi.net/revolutionwifi/2011/03/understanding-wi-fi-carrier-sense.html>
- [13] IEEE. (2012, Mar.) Part 11: Wireless LAN medium Access Control and Physical Layer Specifications. Pdf.
- [14] Poznan University of Technology. invocom. [Online]. [http://www.invocom.et.put.poznan.pl/~invocom/C/P1-4/p1-4\\_en/p1-4\\_7\\_1.htm](http://www.invocom.et.put.poznan.pl/~invocom/C/P1-4/p1-4_en/p1-4_7_1.htm)
- [15] Rafael Estepa Alonso and Antonio Lara Villén, *Apuntes de Redes Multiservicio*. Sevilla, Sevilla, España: Escuela Técnica Superior de Ingeniería, Sep. 2014.
- [16] Wikipedia. (2016, Mar.) wikipedia.org. [Online]. [https://es.wikipedia.org/wiki/IEEE\\_802.11](https://es.wikipedia.org/wiki/IEEE_802.11)
- [18] Rasika Nayanajith. (2014, Oct.) Mrn-Cciw. [Online]. <https://mrncciew.com/2014/10/08/802-11-mgmt-beacon-frame/>
- [17] Kapadia Viral, Sudarshan Patel, and Rutvij Jhaveri, "COMPARATIVE STUDY OF HIDDEN NODE PROBLEM AND SOLUTION USING DIFFERENT TECHNIQUES AND PROTOCOLS," *JOURNAL OF COMPUTING*, , vol. 2, no. 3, p. 68, Feb. 2010.

- [19] Wikipedia. (2015, Oct.) Wikipedia. [Online]. [https://en.wikipedia.org/wiki/IEEE\\_802.11e-2005](https://en.wikipedia.org/wiki/IEEE_802.11e-2005)
- [20] Lucia Lo Bello, Emanuele Toscano, and Vittorio Salvatore, "Capítulo 9," in *A Perspective on the IEEE 802.11e Protocol for the Factory Floor*. Catania, Italia: Universidad de Catania, 2010.
- [21] Cisco. (2011, Apr.) Cisco.com. [Online]. [http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/emob41dg/emob41dg-wrapper/ch5\\_QoS.html](http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/emob41dg/emob41dg-wrapper/ch5_QoS.html)
- [22] Mathilde Benveniste. (2003, Aug.) APS vs Reverse Polling for EDCA. Power Point.
- [23] Nagababu Parsi. (2012, Aug.) IloveWifi.blogspot.com. [Online]. <http://ilovewifi.blogspot.com.es/2012/08/wmm-and-u-apsd-power-save-mode.html>
- [24] Ramapriya Rajamani. (2013, Aug.) riya80211.blogspot. [Online]. <http://riya80211.blogspot.com.es/2013/08/power-save-mode-frames-uapsd.html>
- [25] D-Link. dlink.com. [Online]. <http://www.dlink.com/es/es/support/product/dap-2590-wireless-n-dual-band-poe-access-point>
- [26] [Online]. [http://www.labs.hpe.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.labs.hpe.com/personal/Jean_Tourrilhes/Linux/Tools.html)
- [28] [Online]. <https://support.microsoft.com/en-us/kb/256986>
- [27] Sourceforge. Sourceforge. [Online]. <https://sourceforge.net/projects/ndiswrapper/>
- [29] Bintec. (2016, Feb.) B. [Online]. [https://www.bintec-elmeg.com/portal/downloadcenter/dateien/workshops/current\\_en/ws\\_wlan\\_html\\_en\\_HTML/vowlan\\_infra\\_qos\\_wmm.html](https://www.bintec-elmeg.com/portal/downloadcenter/dateien/workshops/current_en/ws_wlan_html_en_HTML/vowlan_infra_qos_wmm.html)
- [30] IETF. (1994, Feb.) [Online]. [www.ietf.org/rfc/rfc1588.txt](http://www.ietf.org/rfc/rfc1588.txt)
- [31] Wikipedia. www.Wikipedia.org. [Online]. [https://en.wikipedia.org/wiki/Precision\\_Time\\_Protocol](https://en.wikipedia.org/wiki/Precision_Time_Protocol)
- [32] National-Instruments. (2015, Nov.) ni.com. [Online]. <http://www.ni.com/newsletter/50130/en/>
- [33] Wikipedia. (2016, June) Wikipedia. [Online]. [https://en.wikipedia.org/wiki/Precision\\_Time\\_Protocol](https://en.wikipedia.org/wiki/Precision_Time_Protocol)
- [34] [Online]. <https://www ldc.upenn.edu/>
- [35] [Online]. [https://trac.pjsip.org/repos/wiki/Intel\\_IPP\\_Codecs#WindowsInstructions](https://trac.pjsip.org/repos/wiki/Intel_IPP_Codecs#WindowsInstructions)
- [36] [Online]. <https://software.intel.com/en-us/intel-parallel-studio-xe/try-buy>
- [38] [Online]. [https://www.wireshark.org/docs/wsug\\_html\\_chunked/ChCapCaptureFilterSection.html](https://www.wireshark.org/docs/wsug_html_chunked/ChCapCaptureFilterSection.html)
- [37] [Online]. <https://registrationcenter.intel.com/en/forms/?licensetype=2&productid=2489>
- [39] ITU. (2004, May) [Online]. [www.itu.int/rec/T-REC-P.563/es](http://www.itu.int/rec/T-REC-P.563/es)



- 
- [40] ITU. (2005, Nov.) itu.int. [Online]. <http://www.itu.int/rec/T-REC-P.862>
- [41] Wikipedia. (2015, Oct.) wikipedia.org. [Online]. <https://en.wikipedia.org/wiki/PESQ>
- [42] (2015, Nov.) networkingnerd.net. [Online]. [www.networkingnerd.net/tag/mean-opinion-score](http://www.networkingnerd.net/tag/mean-opinion-score)
- [43] ITU-T. (2015, June) The E model: a computational model for use in transmission planing. Pdf.
- [44] Phayung Meesad, Sirapat Boonkrong, and Herwig Unger. (2016, June) Recent Advances in Information and Communication technology 2016. Pdf.
- [45] TamoSoft. [www.tamos.com](http://www.tamos.com). [Online]. [http://www.tamos.com/htmlhelp/voip-analysis/mosandr\\_factor.htm](http://www.tamos.com/htmlhelp/voip-analysis/mosandr_factor.htm)
- [46] [Online]. <https://www.greyscale.com/download/shareware.asp>
- [48] [Online]. <https://www.visualstudio.com/es-es/downloads/download-visual-studio-vs.aspx>
- [47] [Online]. [https://sourceforge.net/projects/ptpd/files/latest/download?source=typ\\_redirect](https://sourceforge.net/projects/ptpd/files/latest/download?source=typ_redirect)
- [49] Rafael Estepa Alonso and Antonio Lara Villén, *Apuntes de Redes Multiservicio*. Sevilla, Sevilla, España: Escuela Técnica Superior de Ingeniería, 2014.
- [50] [Online]. [https://www.wireshark.org/docs/wsug\\_html\\_chunked/ChCapCaptureFilterSection.html](https://www.wireshark.org/docs/wsug_html_chunked/ChCapCaptureFilterSection.html)

# ANEXO A: CONFIGURACIÓN DE LOS EQUIPOS

## A.1 Sincronización de los equipos con PTP

### A.1.1) PCs

En el caso de los ordenadores de sobremesa, estos actuarán como esclavos (slaves) del protocolo PTP, y sincronizarán sus relojes con el del maestro.

Para poder llevar a cabo esta sincronización, se eligió el software Domain Time II de Greyware Automation Products, un software para sincronizar dispositivos en red que soporta múltiples protocolos de sincronización: DT2 (Domain Time II Protocol), NTP (Network Time Protocol), DayTime, LANMan y PTPv2.

La suite de Domain time II proporciona 3 paquetes para instalar: Domain Time II Client, que será el que emplearemos, Domain Time II Server, para configurar el equipo como reloj maestro y Domain Time II Manager para gestionar y monitorizar todos los equipos sincronizados de la red

El ejecutable que contiene el instalador puede obtenerse de la web del desarrollador [38], y puede encontrarse también en el CD adjunto en */Software/dt52eval*, concretamente la versión 5.2.b.20160415. Este producto permite su utilización de forma gratuita durante un periodo de 30 días.

Para instalar el software basta con hacer click sobre el ejecutable y seguir los pasos del asistente de configuración, que al finalizar iniciará la aplicación.

Una vez instalado se debe utilizar el Domain Time Applet para poder configurarlo.

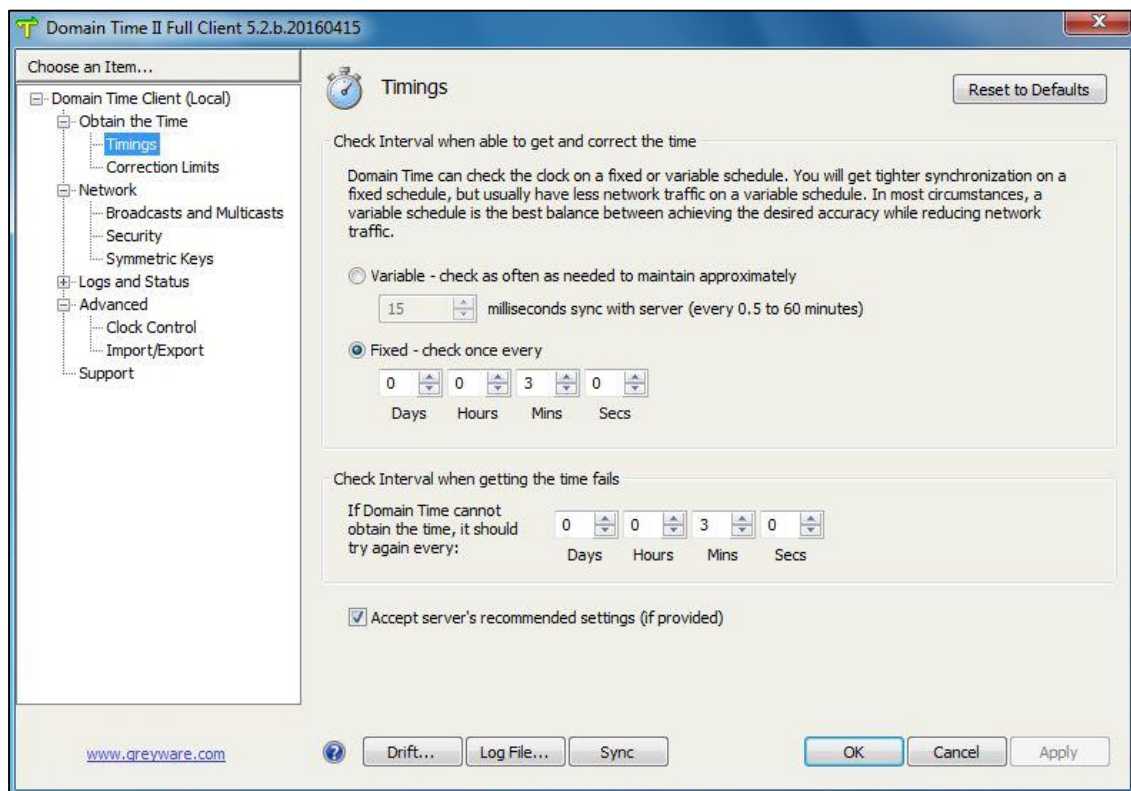


Figura A-1. Detalle del Domain Time Applet

En primer lugar, en la pestaña *Obtain Time* se marca la opción *Set this machine's clock by querying a list of servers*.

Dentro de la lista de servidores de reemplazo (*List of backup time sources to use*) se selecciona al menos un servidor para que si el reloj maestro se desconecta de la red el equipo siga teniendo una fuente de tiempo fiable.

En la pestaña *Timing* se marcan las opciones *Check Interval when able to get and correct the time* y *Check Interval when getting the fails* y se les establece el mismo periodo de tiempo.

Nuevamente se utiliza la pestaña *Obtain Time*, en este caso para marcar la opción *Enable IEEE 1588 Precision Time Protocol (PTPv2)*.

Ahora se debe acceder desde esta misma pestaña a las opciones del protocolo IEEE 1588, y dentro de las mismas se desmarcan las opciones *Require strict IEEE 1588-2008 standard compliance* y *Require ptpTimescale (IEEE 15878-2008 section 8.2.4.8)*. También es necesario modificar el desplegable *PTP Profile*, seleccionando la opción *Always use unicast for delay measurement mechanism*. Por último es necesario desmarcar la opción *Crosscheck with other sources if delta exceeds*.

Tras esto, en la pestaña *Network* se marcan las opciones *Initiate rebind and resync if IP address changes*, *Enumerate multicast interfaces during IPv4/IPv6 bind* y *Reply to multicasts using incoming interface if possible*.

Una vez que se hayan realizado todos los pasos anteriores, el equipo estará listo para sincronizar su reloj con el del maestro. Si no se desea esperar el periodo de sincronización, basta con hacer click en la opción *Synchronize*.

En la figura que se muestra a continuación puede observarse el intercambio de mensajes esperado entre el maestro (10.2.0.5) y uno de sus esclavos. Puede utilizarse como referencia para comprobar que el protocolo funciona.

|    |                 |          |             |       |                       |
|----|-----------------|----------|-------------|-------|-----------------------|
| 85 | 13:18:07.633631 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 106 Announce Message  |
| 86 | 13:18:07.633754 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 86 Sync Message       |
| 87 | 13:18:07.633884 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 86 Follow_Up Message  |
| 88 | 13:18:08.110858 | 10.2.0.2 | 224.0.1.129 | PTPV2 | 86 Delay_Req Message  |
| 89 | 13:18:08.111467 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 96 Delay_Resp Message |
| 90 | 13:18:08.633240 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 86 Sync Message       |
| 91 | 13:18:08.633360 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 86 Follow_Up Message  |
| 92 | 13:18:09.110896 | 10.2.0.2 | 224.0.1.129 | PTPV2 | 86 Delay_Req Message  |
| 93 | 13:18:09.111504 | 10.2.0.5 | 224.0.1.129 | PTPV2 | 96 Delay_Resp Message |

Figura A-2. Intercambio de mensajes PTP de sincronización

#### A.1.2) Equipo Sniffer

Para el PC que actúa de sniffer, que utiliza el sistema operativo Mac OS x El Capitán se utilizó una implementación de código libre de un servidor/cliente PTP, concretamente *Precision Protocol Time Daemon 2*. El código fuente de dicho programa está disponible en SourceForge [45] y puede encontrarse también en el CD adjunto en */Software/ptpd-2.3.1*, concretamente la versión 2.3.1.

Para la instalación de este programa son necesarias la librería *libpcap* y las *GNU Autotools*.

En primer lugar se descomprime el *.tar.gz* que contiene el código fuente, y dentro del directorio que se genera se ejecuta el comando *./configure* que comprueba si se dispone de las librerías y las herramientas necesarias para la configuración. Existen varias opciones que se le pueden indicar a este comando para añadir funcionalidades extra (no son necesarias en este caso) que pueden consultarse en el fichero *INSTALL.txt* del directorio generado tras la descompresión.

Una vez hecho se ejecuta *sudo make*, tras lo que se recomienda realizar un test predefinido. Para realizarlo, basta con ejecutar *./src/ptpd2 -c test/client-e2e-socket.conf* habiendo editado previamente el fichero *test/client-e2e-socket.conf* para añadir la línea *ptpengine:interface = ethXX* siendo *ethXX* el interfaz en el que se escuchan las peticiones de sincronismo y se difunden las tramas del maestro.

Si el test es satisfactorio (Basta con comprobar la salida en */var/run/ptpd2.event.log*), se procede a ejecutar *sudo make install*.

Una vez hecho esto, se ejecuta el maestro con el siguiente comando *sudo ./src/ptpd2 -c ./src/ptpd2.conf.default-full*. El argumento que se le proporciona es un fichero de configuración, que por su complejidad, se proporciona directamente preconfigurado en */Software/Modificaciones/ptpd2.conf.default-full*.

Es importante tener en cuenta que a pesar de que este fichero está preparado para que el maestro arranque y funcione correctamente, es necesario indicar en la línea *ptpengine:interface = ethXX* el interfaz correspondiente que se quiera emplear.

```
MacBook-Pro-de-Miguel:ptpd-2.3.1 miguelfontanilla$ sudo ./src/ptpd2 -c ./src/ptpd2.conf.default-full
Unknown configuration entry: clock:set_rtc_on_step - setting will be ignored
Unknown configuration entry: clock:max_offset_ppm - setting will be ignored
Unknown configuration entry: global:cpuaffinity_cpucore - setting will be ignored
```

Figura A-3. Ejemplo de arranque del servidor ptpd2

Una vez sincronizados los relojes de los equipos que se comunican con el del maestro se realizó una prueba para determinar la desviación media de los relojes de cada equipo con respecto al reloj maestro. Para ello se utilizaron dos redes con diferentes IPs: la red 10.2.0.0/16 empleada para el tráfico de PTP y para la que se utilizaron exclusivamente conexiones Ethernet, y la red 10.1.0.0/16 para el tráfico de datos entre los equipos PC1 Y PC2 mostrados en la figura B-4.

El empleo de Ethernet en la red de sincronismo se debe a que debido a la posibilidad de colisión en un enlace inalámbrico, el cálculo del retardo extremo a extremo que realiza PTP y que se analiza con detalle en el tema 4, puede generar medidas irreales, lo que generaría una mala sincronización.

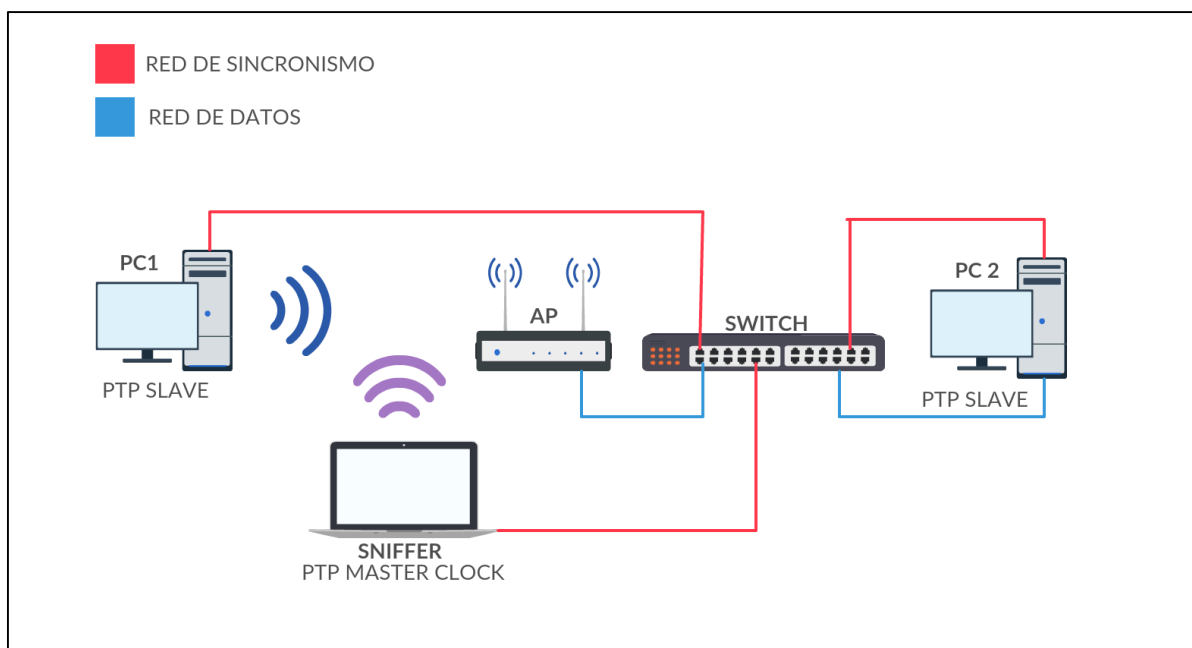


Figura A-4. Esquema de la red utilizada para las pruebas de sincronismo

Para realizar la prueba se lanzó un ping desde el PC1 al PC2 y posteriormente un ping en sentido contrario. Tanto los equipos que intercambiaron el tráfico ICMP como el sniffer realizaron una captura de paquetes en el intervalo en el que se realizó la comunicación. Posteriormente se contrastaron las marcas de tiempo de cada equipo con las ofrecidas por el equipo maestro y se calculó la desviación entre los relojes esclavos y el maestro. Todos estos datos se recogen en un fichero de Excel que se puede encontrar en `/Stats/Ping_PTP` del CD adjunto.

La desviación media obtenida fue de 0,411 ms para el PC1 y de 0,219 ms para el PC2, tiempos relativamente altos para el protocolo PTP, ya que en el documento en el que se define el mismo estándar se especifican desviaciones desde entre 1 a 10 microsegundos hasta niveles de submicrosegundos. No obstante, es necesario tener en cuenta, que empleamos un switch que no está diseñado para el soporte de protocolos de tiempo de alta precisión y el reloj maestro tampoco es un reloj de alta precisión, sino el reloj de un equipo, por lo que consideraremos estos resultados aceptables.

Los resultados obtenidos se contrastaron también con los medidos por el software Domain Time II en cada PC para verificar la exactitud de las medidas.

```

Wait expired after 3 minutes
Summary: 1 sample (aggregate); delta is -0.0000095 seconds, latency +0.0003548
Local clock aligned backward to match 10.2.0.5; variance -0.0000095 seconds, protocol PTPv2, latency 0.35 ms (PTPv2 steering)
Next time check due in 3 minutes (fixed schedule)

Wait expired after 3 minutes
Summary: 1 sample (aggregate); delta is -0.0000105 seconds, latency +0.0003517
Local clock aligned backward to match 10.2.0.5; variance -0.0000105 seconds, protocol PTPv2, latency 0.35 ms (PTPv2 steering)
Next time check due in 3 minutes (fixed schedule)

CPL trigger detected
Summary: 1 sample (aggregate); delta is -0.0000006 seconds, latency +0.0003716
Local clock aligned backward to match 10.2.0.5; variance -0.0000006 seconds, protocol PTPv2, latency 0.37 ms (PTPv2 steering)
Next time check due in 3 minutes (fixed schedule)

```

Figura A-5. Latencia del PC1 medida por Domain Time II

```

Wait expired after 3 minutes
Summary: 1 sample (aggregate); delta is -0.0000076 seconds, latency +0.0002691
Local clock aligned backward to match 10.2.0.5; variance -0.0000076 seconds, protocol PTPv2, latency 0.27 ms (PTPv2 steering)
Next time check due in 3 minutes (fixed schedule)

CPL trigger detected
Summary: 1 sample (aggregate); delta is -0.0000085 seconds, latency +0.0002698
Local clock aligned backward to match 10.2.0.5; variance -0.0000085 seconds, protocol PTPv2, latency 0.27 ms (PTPv2 steering)
Next time check due in 3 minutes (fixed schedule)

CPL trigger detected
Summary: 1 sample (aggregate); delta is -0.0000211 seconds, latency +0.0002862
Local clock aligned backward to match 10.2.0.5; variance -0.0000211 seconds, protocol PTPv2, latency 0.29 ms (PTPv2 steering)
Next time check due in 3 minutes (fixed schedule)

```

Figura A-6. Latencia del PC2 medida por Domain Time II

Puede observarse como los tiempos son similares, por lo que consideraremos los resultados correctos para el escenario planeado.

Es necesario comentar que también se realizaron pruebas para determinar el retardo introducido por el switch, conectando directamente el maestro y el esclavo mediante un cable Ethernet, obteniendo una mejora cercana a la mitad del retardo anteriormente calculado, pero el escenario en este caso no sería escalable, por lo que no se utilizó esta topología.

## A.2 Configuración del AP

Para la configuración del AP DAP-2590 se optó por utilizar el interfaz web debido a la simplicidad del mismo y el fácil acceso que proporciona.

El AP tiene por defecto la dirección IP 192.168.0.50 y el servidor DHCP interno deshabilitado, así que basta con asignar una IP dentro del rango de direcciones posibles bien en el interfaz inalámbrico o en el interfaz Ethernet que se emplee (el AP dispone de una entrada Ethernet).

Una vez hecho esto basta con acceder a la dirección 192.168.0.50 en el navegador y se redirigirá a una página de acceso en la que se solicitan unos credenciales. Se utiliza por defecto el usuario *admin* y no es necesario introducir clave alguna.

Tras acceder se muestra el menú principal: en la parte izquierda se pueden elegir las diferentes categorías de opciones de configuración mediante despleables y en el centro aparecen las opciones y parámetros concretos a modificar.

En el caso de que el AP tenga alguna configuración previa almacenada, puede resetearse manualmente dejando pulsado un botón situado en la parte trasera durante 10 segundos.

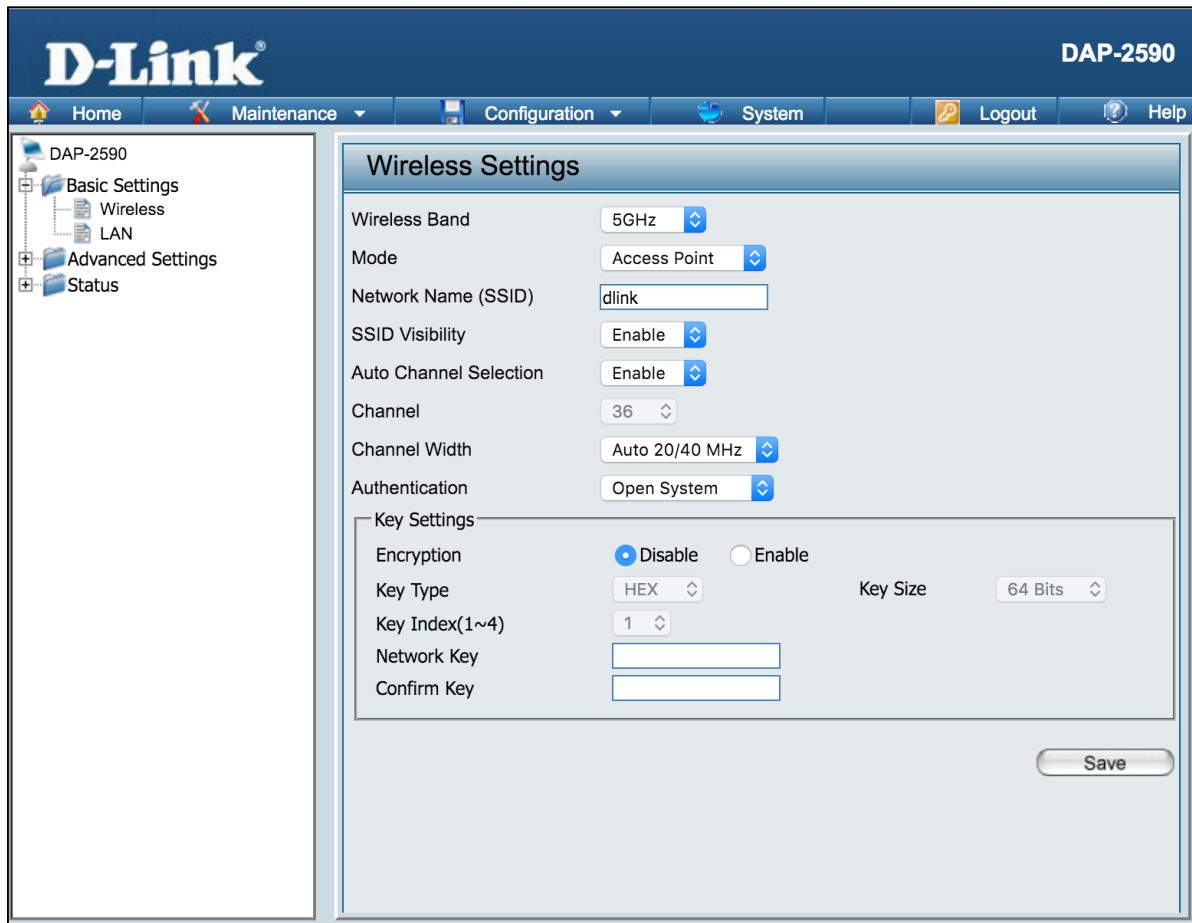


Figura A-7. Interfaz de configuración web del DAP-2590

Las modificaciones que se han realizado con respecto a la configuración por defecto son las siguientes:

En *Basic Settings*>*Wireless* se ha fijado la banda de frecuencia en 5GHz para reducir las interferencias con otras redes inalámbricas. Así mismo se ha seleccionado al ancho de banda canal automático entre 20 MHz y 40 MHz, pulsando el botón *Save* tras realizar los cambios.

En *Advanced Settings*>*Performance* se ha activado el modo WMM, se ha fijado un Beacon Interval de 500 ms, un DTIM Interval de 1, un ACK Timeout de 25  $\mu$ s y el modo wireless Mixed 802.11n, 802.11a, pulsando el botón *Save* tras realizar los cambios.

Una vez realizadas las modificaciones, es necesario pulsar en la ventana *Configuration* de la barra de tareas superior y seleccionar *Save and Activate* para que los cambios introducidos se hagan efectivos en el AP. Tras esto, es necesario esperar unos segundos tras los que se activará la nueva configuración.

Para agilizar este proceso se proporciona en el CD adjunto el fichero de configuración del AP que puede cargarse seleccionando en la barra de tareas superior *Maintenance*>*Configuration File*>*Upload file*. El fichero se puede encontrar en la ruta */Software/config.dcf*.

### A.3 Configuración del Switch

El switch empleado para las pruebas es el HP ProCurve 2610-24, que puede configurarse tanto por interfaz serie como por ssh. Para la configuración mediante el interfaz serie se empleó en programa *putty.exe* con un régimen binario de 9600 baudios. Básicamente se subdividió el switch en 2 VLANs: la de datos y la de sincronismo. Para ello basta con ejecutar los siguientes comandos en el terminal de putty una vez que se establezca la conexión:

```
>config
```

```
>vlan 1
```



```
>name "datos"  
>untagged xx-yy  
>exit  
>vlan 2  
>name "sync"  
>untagged zz-kk  
>exit
```

Donde xx-yy indica el rango de puertos en los que se conectan el interfaz Ethernet del AP y de los interfaces Ethernet para el envío de VoIP de los equipos, mientras que zz-kk indica el rango de puertos en los que se conectan los esclavos y el maestro PTP.

Con esto se evita la difusión de tráfico PTP a través del AP, problema que fue detectado durante la realización de las pruebas.

#### A.4 Configuración de los equipos finales

El software a instalar en los equipos finales, si se desea compilar pjsua en lugar de utilizar la versión modificada que puede encontrarse en */Software/pjsua-mod.exe*, es Visual Studio, en concreto para el proyecto se empleó la versión 2012, que puede descargarse de la web de Microsoft [38].

Para compilar pjsua basta con seguir las instrucciones que se ofrecen en la web de documentación de pjsip, y en el caso de querer incluir nuevos códecs, pueden integrarse los ofrecidos por las *Intel Integrated Performance Primitives*.

Tras la fase de investigación (Capítulo 3), se determinó que el adaptador adecuado para el desarrollo de las pruebas era el Intel AC-7260. A continuación se indican los pasos a seguir para que el equipo en el que se instale funcione con o sin UAPSD.

Para activar UAPSD:

1. Modificar el registro ejecutando el archivo de registro de Windows */Modificaciones/Win7DSCPOverride.REG* del CD adjunto.
2. Modificar las preferencias de energía del sistema accediendo a *Panel de control>Opciones de energía>Cambiar la configuración del plan>Cambiar la configuración avanzada de energía>Configuración de adaptador inalámbrico>modo Ahorro de energía máximo*.
3. Acceder a la Consola de Gestión de Microsoft (Inicio>Ejecutar>MMC) y configurar las políticas de QoS para la aplicación de VoIP correspondiente (Capítulo 3).
4. Acceder al interfaz de configuración de interfaces de red Windows, y activar las opciones de UAPSD y QoS ad-hoc.
5. Seleccionar la protección de modo mixto RTS/CTS o RTS-to-Self.

Para desactivar UAPSD:

1. Acceder al interfaz de configuración de interfaces de red Windows, y desactivar las opciones de UAPSD y QoS ad-hoc.
2. Seleccionar la protección de modo mixto RTS/CTS o RTS-to-Self.

# ANEXO B: SCRIPTS Y CÓDIGO FUENTE

## B.1 autcapture.sh

Se trata de un script de bash diseñado para facilitar la automatización de las capturas de paquetes en el equipo sniffer.

Se encarga de ir capturando cada llamada en un fichero independiente que posteriormente se irá comprimiendo con el objetivo de minimizar el espacio ocupado.

Los parámetros principales que emplea son:

- **FILTER**, una expresión para el filtrado que sigue la sintaxis de libpcap.
- **INTERVAL**, la duración en segundos de cada captura independiente.

El script almacenará las capturas en una carpeta llamada PCAP, que situará en el mismo directorio en el que se encuentre, asignándole el nombre UAPSD.pcap, que posteriormente comprimirá en un fichero llamado UAPSDX.zip, donde X representa el valor de Nfpp.

Debe tenerse en cuenta, que por defecto se realizan 10 capturas con cada ejecución, por lo que para códecs como iLBC, cuyo máximo valor posible de Nfpp es 7, se debe modificar la opción de tshark

*-a duration duration:\$((INTERVAL\*Nfpp))*

siendo Nfpp el valor máximo que alcanza para el códec.

Este script no recibe parámetros de entrada.

## B.2 UAPSD.c

Es un programa escrito en C que procesa las capturas de Wireshark una vez que estas han sido volcadas a texto plano. Es muy importante que se exporten las capturas a texto plano utilizando la opción de Wireshark *File>Export Packet Dissections>As Plain Text* sin marcar la casilla *Details* y marcando la casilla *Bytes*, o de lo contrario el programa no podrá interpretar los resultados.

Se trata del programa más complejo desarrollado, tanto por extensión (más de 6300 líneas de código), como por funcionalidad, por ello para entender su funcionamiento lo dividiremos por bloques; un primer bloque de búsqueda e identificación de los paquetes y un segundo bloque de procesamiento de los datos del mismo. Realmente el programa también se subdivide internamente en función de si el paquete es ascendente o descendente.

En primer lugar es necesario señalar que el funcionamiento se repite para cada códec de forma similar, salvo por alguna modificación necesaria para garantizar el correcto funcionamiento, como por ejemplo el uso de la cabecera 802.11 o radiotap, o la gestión de errores específicos de una captura.

El programa tiene dos modos de funcionamiento: *uapsd*, y *nouapsd*. La elección de uno u otro modifica la forma en la que se gestionan las estadísticas, pero no la forma en la que se desarrolla la ejecución del programa. Básicamente, el programa recorre línea a línea buscando una determinada estructura que le permita saber si es el comienzo de un paquete o no. Esto se consigue gracias a las líneas que Wireshark inserta con el siguiente formato para cada paquete:

| <i>No.</i> | <i>Time</i> | <i>Source</i> | <i>Destination</i> | <i>Protocol Info</i> | <i>Length</i> |
|------------|-------------|---------------|--------------------|----------------------|---------------|
|------------|-------------|---------------|--------------------|----------------------|---------------|

Una vez encontrada, se leen los distintos parámetros del paquete (origen, destino, tiempo, etc) y se ajustan si fuera necesario en función del protocolo. Cuando se conocen los datos que identifican al paquete, se realiza una secuencia de comprobaciones para determinar que tipo de paquete es (*ACK*, *CTS*, *RTS*, *Datos RTP*, etc), y en el momento en el que una de las comprobaciones sea positiva, se pasa a gestionar el paquete según corresponda; bien en modo *uapsd* o en modo *nouapsd*.

El modo *nouapsd*, que es más sencillo computa el tiempo en idle desde el último paquete transmitido o recibido hasta el actual y posteriormente computa el tiempo que se ha tardado en transmitirlo o recibirlo,



almacenando todos ellos en un acumulador.

El modo `uapsd`, más complejo, utiliza una serie de variables para identificar el estado del algoritmo (explicado en el capítulo 5) en el que se encuentra; `status` y `statXphase` que indican respectivamente el estado y la fase del estado en la que quedó el equipo tras la transmisión o recepción del último paquete.

En ambos modos se calculan las pérdidas a nivel de aplicación para RTP, y las ráfagas de pérdidas observando los números de secuencia de los paquetes. También se obtiene el delay y el jitter a partir de los timestamps de los paquetes RTP y de el tiempo de llegada medido.

El código fuente se encuentra en el CD adjunto en la ruta `/Software/UAPSDC.c`, y se invoca con la siguiente sintaxis:

*UAPSD Fichero Nfpp Códec Modo*

- Fichero es el nombre del fichero de texto plano en el que se ha volcado el fichero de capturas de Wireshark.
- Nfpp indica en número de tramas por paquete empleado, que puede ir de 1 a 10, salvo en el caso del códec iLBC, que sólo llega hasta 7
- Códec indica el códec empleado (`pcma,ilbc,speex`)
- Modo indica si se emplea el modo de ahorro de energía (`uapsd`) o no (`nouapsd`)

Los parámetros obtenidos tras la ejecución se analizan en el capítulo 5.

### **B.3 MOS\_Processing.sh**

Se trata de un script de bash diseñado para realizar de forma automática la evaluación de la calidad de servicio con el software ofrecido por la ITU para la norma P.563, que puede encontrarse en el CD adjunto en la ruta `/Software/T-REC-P.563-200405-I!!SOFT-ZST-E`.

Este script realiza en primer lugar un recorte de las muestras que se desean emplear utilizando el software Sox que puede encontrarse en el CD adjunto en la ruta `/Software/sox-14.4.2` y posteriormente ejecuta el análisis de las muestras con `p563`.

Tras la ejecución, los resultados pueden consultarse en el fichero `out_p563.csv`. Las salidas se generan en formato `.csv` para poder exportarlas con facilidad a Excel y generar las gráficas posteriormente.

Los parámetros principales que emplea son:

- MINS1,MINS2,MINS3 y MINS4, cadenas de texto que contienen el minuto en el que se desea iniciar el recorte de audio y la duración del mismo. MINS1 y MINS2 se utilizan para el audio recogido en el PC1, mientras que MINS3 y MINS4 para el audio grabado PC2.
- CSVFILE, la ruta para el fichero csv generado tras la ejecución del script.
- RUTA, almacena la ruta del ejecutable.
- EXEC, almacena el nombre del ejecutable.
- SOX, almacena la ruta de la herramienta de audio empleada para el recorte.

Es importante señalar que el script espera encontrar las grabaciones organizadas de la siguiente manera:

*Carpeta\_contenedora/pcx/CODEC\_xxUAPSD/OUTPUTX.WAV*

donde `pcx` indica el pc en el que se recogió la grabación, `CODEC` el códec que se empleó, `XX` puede estar vacío (se emplea `UAPSD`) o ser `NO` (no se emplea `UAPSD`), y `OUTPUTX.WAV` es el fichero de audio grabado en el extremo donde `X` indica el valor de `Nfpp`.

Este script recibe un único parámetro de entrada que indica el audio del equipo que se desea emplear: `pc1,pc2` o ambos (`both`).

### **B.4 MOS\_Processing\_2.sh**

Se trata de un script de bash diseñado para realizar de forma automática la evaluación de la calidad de servicio con el software ofrecido por la ITU para la norma P.862, que puede encontrarse en el CD adjunto en la ruta

*/Software/T-REC-P.862-200511-1!Amd2!SOFT-ZST-E.*

Este script realiza en primer lugar un recorte de muestras, tanto originales como degradadas que se desean emplear utilizando el software Sox que puede encontrarse en el CD adjunto en la ruta */Software/sox-14.4.2* y posteriormente ejecuta el análisis de las muestras con p862.

Es necesario señalar que el recorte de las referencias originales se realiza en el apartado *REFERENCE SAMPLES*, donde se pueden indicar el minuto de inicio y la duración.

Tras la ejecución, los resultados pueden consultarse en el fichero *out\_p862.csv*. Las salidas se generan en formato *.csv* para poder exportarlas con facilidad a Excel y generar las gráficas posteriormente.

Los parámetros principales que emplea son:

- MINS1,MINS2,MINS3 y MINS4, cadenas de texto que contienen el minuto en el que se desea iniciar el recorte de audio y la duración del mismo para las muestras degradadas. MINS1 y MINS2 se utilizan para el audio recogido en el PC1, mientras que MINS3 y MINS4 para el audio grabado PC2.
- CSVFILE, la ruta para el fichero csv generado tras la ejecución del script.
- RUTA, almacena la ruta del ejecutable.
- EXEC, almacena el nombre del ejecutable.
- SOX, almacena la ruta de la herramienta de audio empleada para el recorte.
- REFERENCE1,REFERENCE2,REFERENCE3 y REFERENCE4, la ruta de las referencias originales de audio con las que comparar los audios degradados.

Es importante señalar que el script espera encontrar las grabaciones organizadas de la siguiente manera:

*Carpeta\_contenedora/pcx/CODEC\_xxUAPSD/OUTPUTX.WAV*

donde *pcx* indica el pc en el que se recogió la grabación, *CODEC* el códec que se empleó, *XX* puede estar vacío (se emplea *UAPSD*) o ser *NO* (no se emplea *UAPSD*), y *OUTPUTX.WAV* es el fichero de audio grabado en el extremo donde *X* indica el valor de *Nfpp*.

Este script recibe un único parámetro de entrada que indica el audio del equipo que se desea emplear: *pc1*,*pc2* o ambos (*both*).

### **B.5 pjsip\_g.ps1**

Se trata de un script de Windows PowerShell que se encarga de automatizar las llamadas entre los PCs empleados en los escenarios de prueba.

Realmente existen dos versiones; la primera puede encontrarse en el CD adjunto en la ruta */Software/PC1/pjsip\_g.ps1* y la segunda en la ruta */Software/PC2/pjsip\_g.ps1*. La diferencia principal entre ambas es que la segunda está desarrollada para el PC que inicia la llamada mientras que la primera para el PC que la recibe.

Está desarrollado para ejecutarse en el escritorio, concretamente para el usuario *PC-RM1*, pero basta con modificar la línea

*C:\Users\PC1-RM\Desktop\*

Indicando la ruta deseada para su ejecución.

El parámetro más importante de este script es *--config-file*, donde se indica el fichero de configuración a utilizar en cada caso. Se proporcionan tres ficheros de configuración, que pueden encontrarse en la ruta */Software/config/PCX/* del CD *config\_pcma.txt*, *config\_ilbc.txt*, *config\_speex.txt*.

Para que el script funcione es necesario acompañarlo de la carpeta *NFPPS* que puede encontrarse en la misma ruta y con el ejecutable de *pjsua* compilado *pjsua-mod.exe*, también situado en la ruta anterior.

El script genera en el directorio en el que se ejecuta una serie de grabaciones, que contienen el audio recibido en cada extremo de la conversación. Estas grabaciones generadas, toman el nombre *OUTPUTX.WAV* siendo *X* el valor de *Nfpp* en la prueba concreta.

## B.6 pjsua-mod.exe

Se trata de un ejecutable de Windows generado a partir del código fuente de pjsua con Visual Studio. La modificación que se ha realizado en el código fuente consiste en forzar al programa a que lea el valor de Nfpp a utilizar de un fichero de texto llamado NFPP.TXT.

La modificación realizada es mínima, y consiste en añadir las siguientes líneas al fichero stream.c que puede encontrarse en el CD adjunto en */Software/pjproject-2.5.1/pjmedia/src/pjmedia/* en la línea 2087:

```
FILE *fp;
int Nfpp;
fp=fopen("NFPP.TXT","r");
if (fp != NULL)
{
    fscanf(fp,"%d",&Nfpp);
    fclose(fp);
    if (Nfpp > 0)
    {
        stream->codec_param.setting.frm_per_pkt = (pj_uint8_t) Nfpp;
    }
}
```

Los parámetros que recibe este ejecutable son:

- --rec-file=, que indica el nombre del fichero en el que se almacenará el audio de la llamada recibida.
- --auto-rec, que inicia la grabación en el momento en el que comienza la llamada.
- --config-file, que indica el fichero de configuración a utilizar y en el que se encuentran el resto de parámetros que recibe el programa.
- --dis-codec=, que permite deshabilitar el códec especificado.
- --play-file=, que permite indicar el fichero que se empleará como fuente de audio para la llamada.
- --auto-play, que inicia la reproducción del audio en el momento en el que comienza la llamada.
- --duration=, especifica la duración de la llamada.

# GLOSARIO

|   |    |
|---|----|
| VoIP: Voice Over Internet Protocol .....                            | 1  |
| PSTN: Public Switched Telephone Network.....                        | 1  |
| IEEE: Institute of Electrical and Electronic Engineers.....         | 2  |
| WLAN: Wireless Local Area Network .....                             | 2  |
| VoWLAN: Voice over WLAN.....  | 2  |
| UDP: User Datagram Protocol.....                                    | 4  |
| SIP: Session Initiation Protocol.....                               | 7  |
| DNS: Domain Name System .....                                       | 7  |
| HTTP: Hypertext Transport Protocol .....                            | 7  |
| RTP: Real-Time Transport Protocol.....                              | 8  |
| QoS: Quality of Service.....  | 8  |
| RTCP: RTP Control Protocol.....                                     | 8  |
| AP: Access Point.....   | 10 |
| PLCP: Physical Layer Convergence Protocol .....                     | 10 |
| PMD: Physical Medium Dependent .....                                | 10 |
| BSS: Basic Service Set .....  | 11 |
| ESS: Extended service set .....                                     | 11 |
| DCF: Distributed Coordination Function .....                        | 13 |
| PCF: Point Coordination Function.....                               | 13 |
| CSMA/CA: Carrier Sense Multiple Acces with Collision Avoidance..... | 14 |
| MAC: Medium Access Control .....                                    | 14 |
| SIFS: Short Inter Frame Space.....                                  | 14 |
| DIFS: DCF Inter Frame Space.....                                    | 14 |
| PIFS: PCF Inter Frame Space .....                                   | 14 |
| EIFS: Extended Inter Frame Space.....                               | 15 |
| Ts: Time Slot .....   | 15 |
| ACK: Acknowledgement .....  | 15 |
| CW: Contention Window .....   | 15 |
| RTS: Request To Send .....  | 16 |
| CTS: Clear To Send.....   | 16 |
| CW: Contention Window.....  | 16 |
| TIM: Traffic Indication Map.....                                    | 17 |
| DTIM: Delivery Traffic Indication Map.....                          | 17 |

---

|  |    |
|--|----|
| EDCA: Enhanced Distributed Chanel Access.....          | 18 |
| HCCA: Hybrid Coordination Function Chanel Access.....  | 18 |
| TXOP: Transmission Oportunity .....                    | 19 |
| WMM: Wi-Fi Multi Media .....                           | 20 |
| WMM-PS: Wi-Fi Multi Media Power Save .....             | 20 |
| APSD: Automatic Power Save Delivery .....              | 21 |
| U-APSD: Unscheduled Automatic Power Save Delivery..... | 21 |
| EOSP: End Of Service Period .....                      | 21 |
| PTP: Precission Time Protocol.....                     | 45 |
| PESQ: Perceptual Evaluation of Speech Quality .....    | 64 |
| MOS: Mean Opinion Score .....                          | 65 |