



César Solera Calvo

Trabajo Fin de Carrera
Consultor : José López Vicario
Correo electrónico: csolerac@uoc.edu

Evaluación del rendimiento de una red WiFi transportando tráfico VoIP

Trabajo Fin de Carrera

14 de junio del 2014

Índice

1.- INTRODUCCIÓN	2
2.- DESCRIPCIÓN DE VOIP Y CODECS DE AUDIO	3
2.1.- PROTOCOLOS DE VoIP	3
2.2.- ELEMENTOS DE VoIP	6
2.3.- DESCRIPCIÓN DE LOS CODECS	6
2.4.- CÓDEC G.711	8
2.5.- CÓDEC G.723.1	9
2.5.1- DESCRIPCIÓN DE LA PARTE CODIFICADORA	9
2.5.2- DESCRIPCIÓN DE LA PARTE DECODIFICADORA	11
2.6.- CÓDEC G.729	11
2.6.1- DESCRIPCIÓN DE LA PARTE CODIFICADORA	12
2.6.1- DESCRIPCIÓN DE LA PARTE DECODIFICADORA	13
2.7.- RESUMEN	14
3.- DESCRIPCIÓN DE LOS PROTOCOLOS DE WIRELESS 802.11	16
3.1.- ESTÁNDAR 802.11A	17
3.2.- ESTÁNDAR 802.11B	17
3.3.- ESTÁNDAR 802.11G	18
3.4.- ESTÁNDAR 802.11N	19
3.5.- ESTÁNDAR 802.11E	22
4.- PRESENTACIÓN DEL MODELO TEÓRICO PARA LA SIMULACIÓN	26
4.1.- INTRODUCCIÓN	26
4.2.- PARÁMETROS DE LA SIMULACIÓN	26
4.2.1- E-MODEL	26
4.2.2- MOS (MEAN OPINION SCORE)	27
4.2.3- RELACIÓN ENTRE E-MODEL Y MOS	28
4.2.4- PARÁMETROS PARA EL ESTÁNDAR 802.11	28
4.3.- CÁLCULO DEL NÚMERO DE LLAMADAS	28
4.4.- HERRAMIENTA USADA PARA LA SIMULACIÓN	29
5.- RESULTADOS OBTENIDOS SEGÚN EL MODELO	32
5.1.- INTRODUCCIÓN	32
5.2.- CÁLCULO DE MOS EN FUNCIÓN DEL RETARDO	32
5.3.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO DCF/PCF	32
5.4.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA	33
5.5.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA VARIANDO EL TIEMPO DE CP	34
5.6.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS SEGÚN LA VELOCIDAD DE TRANSMISIÓN	35
5.7.- ACLARACIONES DEL APARTADO	36
6.- RESULTADOS OBTENIDOS CON LA AMPLIACIÓN DEL MODELO	37
6.1.- INTRODUCCIÓN	37
6.2.- CÁLCULO DE MOS EN FUNCIÓN DEL RETARDO	37
6.3.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO DCF/PCF	38
6.4.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA	39
6.5.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA VARIANDO EL TIEMPO DE CP	39
6.6.- CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS SEGÚN LA VELOCIDAD DE TRANSMISIÓN	40
7.- CONCLUSIONES	43
8.- MATERIAL DE REFERENCIA	45



ANEXO I - CÓDIGO DE MATLAB DE LA SIMULACIÓN	47
ANEXO II - CÓDIGO DE MATLAB DE LA SIMULACIÓN AMPLIADA	54

Índice de Tablas

TABLA 1 - RESUMEN DE CODECS DE VOZ	8
TABLA 2 - RESUMEN DE CARACTERÍSTICAS DE ANCHO DE BANDA DE LOS CODECS	14
TABLA 3 - RESUMEN DE FAMILIA DE PROTOCOLOS 802.11	16
TABLA 4 - CANALES USADOS POR 802.11B	18
TABLA 5 - CANALES USADOS POR 802.11 B/G	19
TABLA 6 - VELOCIDADES DE 802.11N SEGÚN ÍNDICE MCS	20
TABLA 7 - USO DEL ANCHO DE BANDA DE 40MHZ EN LA BANDA DE LOS 2,4 GHZ	21
TABLA 8 - VALORES DE DIFS SEGÚN LA NORMA USADA	22
TABLA 9 - VALORES DE SIFS SEGÚN LA NORMA USADA	22
TABLA 10 - VALORES DE PIFS SEGÚN LA NORMA USADA	23
TABLA 11 - CÁLCULO DE LOS LÍMITES DEL PARÁMETRO CW	24
TABLA 12 - VALORES POR DEFECTO DE EDCA PARA CADA AC	24
TABLA 13 - EJEMPLO DE PUNTUACIONES MOS DE ALGUNOS CODECS	27
TABLA 14 - VALORES DE MOS Y FACTOR R PARA LA CALIDAD DE TRANSMISIÓN DE VOZ	28
TABLA 15 - PARÁMETROS USADOS PARA EL MODELADO DEL SISTEMA	28
TABLA 16 - PARÁMETROS DE LOS CODECS PARA RETRASOS Y TASAS DE TRANSFERENCIA	29
TABLA 17 - NÚMERO MÁXIMO DE LLAMADAS PARA LOS CODECS ESTÁNDAR EN FUNCIÓN DE LA TASA DE TRANSMISIÓN Y USANDO EL MODO DE FUNCIONAMIENTO DCF/PCF	36
TABLA 18 - NÚMERO MÁXIMO DE LLAMADAS PARA LOS CODECS ESTÁNDAR EN FUNCIÓN DE LA TASA DE TRANSMISIÓN Y USANDO EL MODO DE FUNCIONAMIENTO EDCA/HCCA	36
TABLA 19 - NÚMERO MÁXIMO DE LLAMADAS PARA LOS CODECS ESTÁNDAR EN FUNCIÓN DE LA TASA DE TRANSMISIÓN Y USANDO EL MODO DE FUNCIONAMIENTO DCF/PCF	41
TABLA 20 - NÚMERO MÁXIMO DE LLAMADAS PARA LOS CODECS ESTÁNDAR EN FUNCIÓN DE LA TASA DE TRANSMISIÓN Y USANDO EL MODO DE FUNCIONAMIENTO EDCA/HCCA	41

Índice de Ilustraciones

ILUSTRACIÓN 1 - EJEMPLO DE DATAGRAMA RTP	5
ILUSTRACIÓN 2 - DIAGRAMA DE BLOQUES DEL CODIFICADOR G.723.1	11
ILUSTRACIÓN 3 - DIAGRAMA DE BLOQUES DEL DECODIFICADOR G.723.1	11
ILUSTRACIÓN 4 - DIAGRAMA DE BLOQUES DEL CODIFICADOR G.729	13
ILUSTRACIÓN 5 - DIAGRAMA DE BLOQUES DEL DECODIFICADOR DE G.729	14
ILUSTRACIÓN 6 - SOLAPAMIENTO DE CANALES EN LA BANDA DE 2,4 GHZ	18
ILUSTRACIÓN 7 - CANALES SIN SOLAPAMIENTO DE 802.11 B/G/N EN LOS 2,4 GHZ	21
ILUSTRACIÓN 8 - ENCAJE DE 802.11 DENTRO DEL MODELO OSI	23
ILUSTRACIÓN 9 - EJEMPLO DE GENERACIÓN DE LOS CAPS DURANTE LOS CP	24
ILUSTRACIÓN 10 - EJEMPLO DE RED WIFI PARA VOIP	26
ILUSTRACIÓN 11 - MOS DE CODECS TÍPICOS DE VOIP EN FUNCIÓN DEL RETARDO DE LA RED	32
ILUSTRACIÓN 12 - NÚMERO MÁXIMO DE LLAMADAS SOPORTADAS EN DCF/PCF	33
ILUSTRACIÓN 13 - NÚMERO MÁXIMO DE LLAMADAS EN EL MODO EDCA/HCCA	34
ILUSTRACIÓN 14 - NÚMERO MÁXIMO DE LLAMADAS QUE SE PUEDEN REALIZAR MIENTRAS SE INCREMENTA EL TIEMPO DE CP	34
ILUSTRACIÓN 15 - NÚMERO MÁXIMO DE LLAMADAS SOPORTADO EN MODO DCF/PCF EN FUNCIÓN DE LA TASA DE TRANSFERENCIA	35
ILUSTRACIÓN 16 - NÚMERO MÁXIMO DE LLAMADAS SOPORTADO EN MODO EDCA/HCCA EN FUNCIÓN DE LA TASA DE TRANSFERENCIA	35
ILUSTRACIÓN 17 - MOS DE LOS CODECS ESTÁNDAR EN FUNCIÓN DEL RETARDO	38
ILUSTRACIÓN 18 - NÚMERO DE LLAMADAS MÁXIMAS EN FUNCIÓN DEL TIEMPO DE SF EN MODO DCF/PCF	38
ILUSTRACIÓN 19 - NÚMERO DE LLAMADAS MÁXIMAS EN FUNCIÓN DEL TIEMPO DE SF EN MODO EDCA/HCCA	39
ILUSTRACIÓN 20 - NÚMERO MÁXIMO DE LLAMADAS MIENTRAS SE INCREMENTA EL TIEMPO DE CP EN EL MODO EDCA/HCCA	40
ILUSTRACIÓN 21 - NÚMERO MÁXIMO DE LLAMADAS SOPORTADO EN DCF/PCF CON SF=120MS EN FUNCIÓN DEL INCREMENTO DE LA VELOCIDAD DE TRANSMISIÓN	40
ILUSTRACIÓN 22 - NÚMERO MÁXIMO DE LLAMADAS SOPORTADO EN EDCA/HCCA CON SF=120MS EN FUNCIÓN DEL INCREMENTO DE LA VELOCIDAD DE TRANSMISIÓN	41

Índice de Ecuaciones

ECUACIÓN 1 - CÁLCULO DE TIEMPO DIFS	22
ECUACIÓN 2 - CÁLCULO DEL TIEMPO PIFS	23
ECUACIÓN 3 - CÁLCULO DEL TIEMPO AIFS	24
ECUACIÓN 4 - CÁLCULO DEL FACTOR R	27
ECUACIÓN 5 - TRADUCCIÓN DE FACTOR R A ESCALA MOS	28
ECUACIÓN 6 - ECUACIÓN PARA EL CÁLCULO DE $T_{\text{POLL-CP}}$	29
ECUACIÓN 7 - ECUACIÓN PARA EL CÁLCULO DE $T_{\text{POLL-CP}}$	29
ECUACIÓN 8 - ECUACIÓN PARA EL CÁLCULO DE T_{OVERHEAD}	29
ECUACIÓN 9 - ECUACIÓN PARA EL CÁLCULO DEL TOTAL DE LLAMADAS EN UN PERIODO DE CFP	29
ECUACIÓN 10 - ECUACIÓN PARA EL CÁLCULO DEL TOTAL DE LLAMADAS EN UN PERIODO DE CP	29
ECUACIÓN 11 - CÁLCULO DEL TIEMPO DE CFP PARA EL MODO DCF/PCF	33

1.- INTRODUCCIÓN

En el presente Trabajo Fin de Carrera (en adelante TFC), se pretende dar una visión del rendimiento que tiene la VoIP (*Voice over IP*) en las redes WLAN. Para ello se va a usar como referencia el artículo “*Capacity Evaluation of VoIP in IEEE 802.11e WLAN Environment*”¹ y así poder realizar una simulación del modelo que propone para comprobar los resultados. Posteriormente se realizarán cambios en el mismo para comprobar cómo se comporta dicho modelo. Estas modificaciones estarán basadas en los cambios que han sufrido los protocolos de wireless, ya que se han ampliado las partes de QoS (*Quality of Service*) y los anchos de banda que se usan, incrementando así el número de canales que se pueden usar en un entorno de wireless.

Se va a intentar dar un conocimiento global de qué es la VoIP y su rápida aceptación en el mundo así como qué codecs de audio usa para poder transmitir la voz a través de una conexión IP. Estos codecs son los responsables de la compresión de la voz para poder transmitir un mayor número de información en el menor ancho de banda posible sin sufrir una pérdida en la calidad de la voz.

A continuación se dará una visión de qué es el estándar 802.11 y cómo ha ido evolucionando. Se hará un especial hincapié en la definición de 802.11e ya que en él se basa el uso de QoS en redes wireless. Esta definición es la que va a permitir asegurar el número de canales de voz sobre el ancho de banda de la red sin que se produzcan solapamientos o efectos como la voz entrecortada debida a la pérdida de paquetes o retrasos en los mismos debido a efectos negativos como el de la saturación.

Una vez entendido el entorno en el que se van a desarrollar las pruebas, se procederá a presentar en qué se basa el modelo que propone el autor del artículo para después intentar reproducirlo con una herramienta estilo Matlab. Los resultados obtenidos serán analizados para ver si se ha conseguido este objetivo. A continuación se harán las modificaciones pertinentes al mismo para realizar de nuevo una simulación y obtener nuevos resultados para poder compararlos con el original.

Finalmente se realizará una conclusión para extraer toda la información posible de las simulaciones realizadas y conseguir un análisis que nos aporte información útil para aplicaciones reales dentro del ámbito de las comunicaciones de voz sobre redes wireless.

¹ <http://www-sop.inria.fr/planete/atrad/paper-ccnc06-trad.pdf>

2.- DESCRIPCIÓN DE VOIP Y CODECS DE AUDIO

La VoIP es una tecnología que nos permite enviar señales de voz de forma digital a través de redes de comunicación de datos como la red IP. Esto es una gran diferencia a la telefonía tradicional que envía la voz a través de la red conmutada tradicional (PSTN en sus siglas en inglés), ya que se deja de usar una red de conmutación de circuitos con camino dedicado a cada conversación para pasar a usar una red de conmutación de paquetes de propósito general con elección de camino para cada transmisión de datos. A primera vista esto puede parecer perjudicial para la transmisión de voz, pero debido a los protocolos que se usan en las redes IP y las velocidades de transmisión, se compensa el no tener un camino dedicado y que las llamadas no sufran mucho retardo.

Para esta transmisión de voz, el protocolo de red que usa la VoIP es el UDP. La elección de este protocolo es la rapidez con la que manda la información comparado con el protocolo TCP. Es cierto que UDP no asegura la entrega de los paquetes de datos en el otro extremo de la comunicación, pero debido a la fiabilidad de las redes actuales, esto no supone un gran problema ya que el porcentaje de paquetes que se pueden perder es muy pequeño. Es cierto que TCP asegura la entrega de los paquetes, pero al tratarse de una aplicación en tiempo real, lo que se necesita es mucha rapidez para que no se produzcan retardos, y la comunicación sea lo más parecida posible a una llamada por la red telefónica convencional. Este hecho es el determinante para que el protocolo sobre el que se envían las comunicaciones de VoIP sea UDP.

Por otro lado, TCP es un protocolo mucho más fiable, en donde lo que se busca es que los paquetes se entreguen en su totalidad para no perder información. Un ejemplo de aplicación que usa TCP es el protocolo HTTP. En este caso, lo que interesa es que toda la información de una página web se entregue en el destino con total fiabilidad para que el destinatario no pierda nada de información. Aquí da un poco igual si los paquetes se entregan más o menos rápido, lo fundamental es que se entreguen. Por ello se usa TCP y no UDP a pesar de que se pueda pensar que lo que prima es la velocidad.

2.1.- PROTOCOLOS DE VoIP

En los protocolos para VoIP, se conocen las comunicaciones por los protocolos que se usan para el envío de la señalización, ya que el envío de la voz propiamente dicho se realiza con el uso de algún otro protocolo. Con esta premisa, los protocolos más importantes de VoIP son los siguientes:

- H.323: Es el protocolo más antiguo usado para comunicaciones de VoIP. Fue definido por el ITU-T en noviembre de 1993. LA señalización que transporta se basa en el protocolo de comunicaciones Q.931, que es el que usan las redes RDSI. Fue el primer protocolo que hizo uso de RTP (*Real-time Transport Protocol*) de IETF (*Internet Engineering Task Force*) para la transmisión de audio y vídeo.
- SIP (*Session Initiation Protocol*): Es el protocolo más extendido actualmente debido a sus implementaciones en centralitas (PBX) open source como Asterisk. Fue definido en 1996 y en noviembre del año 2000 se acepta como 3GPP del subsistema multimedia de IP. Se encarga del manejo de la señalización de una comunicación VoIP. Las funciones que implementa para la señalización son muy similares a las del protocolo SS7 (*Signaling System 7*) que es el protocolo que implementan las redes de telefonía actuales con funciones de inteligencia de red. La parte del envío de la propia voz se realiza usando el protocolo RTP.

La mayor diferencia entre estos dos protocolos es que el primero viene definido por empresas de telecomunicaciones, mientras que el segundo surge, directamente, de la comunidad de usuarios de tecnología IP. Este hecho hace que el segundo sea más aceptado que el primero dentro de la pila de protocolos TCP/IP.

El motivo por el que estos protocolos son los más usados para la VoIP es que H.323 fue el primer protocolo que se implementó para este tipo de comunicaciones, con lo que bastantes fabricantes lo adoptaron para sus centralitas PBX. Un ejemplo de ello es Avaya. El segundo se ha extendido muy rápidamente por ser una implementación sencilla y que es fácilmente adaptable a los dispositivos de usuario final. Además, tiene la opción de poder usar softphones, terminales software que se pueden ejecutar en un ordenador o en los actuales smartphones y tablets.

Además, el hecho del desarrollo de la centralita *Open Source Asterisk* ha impulsado mucho el uso de SIP ya que se puede disponer de una centralita PBX con una inversión pequeña de dinero. Se han realizado muchos desarrollos sobre Asterisk para impulsar el uso de la VoIP como los *software* de FreePBX o Elastix entre otros. Como muchos fabricantes han apostado por este tipo de tecnología, es muy fácil encontrar casi cualquier tipo de dispositivo que sea integrable con tecnología Asterisk ya que hace uso del protocolo SIP.

Y aunque parezca algo inverosímil, Asterisk tiene su propio protocolo de comunicaciones que es IAX (*Inter-Asterisk eXchange protocol*), el protocolo que ha triunfado, incluso usando esta centralita *Open Source*, ha sido SIP. El motivo es que SIP fue desarrollado antes y está estandarizado, mientras que IAX está pendiente de estandarización, y por ese motivo no lo implementan muchos fabricantes.

Actualmente IAX está en su segunda versión y se le conoce comúnmente como IAX2. Las principales diferencias entre SIP e IAX son:

- Ancho de banda.
IAX utiliza un menor ancho de banda que SIP ya que los mensajes son codificados de forma binaria mientras que en SIP son mensajes de texto. Asimismo, IAX intenta reducir al máximo la información de las cabeceras de los mensajes reduciendo también el ancho de banda.
- NAT
En IAX la señalización y los datos viajan conjuntamente con lo cual se evitan los problemas de NAT que frecuentemente aparecen en SIP. En SIP la señalización y los datos viajan de manera separada y por eso aparecen problemas de NAT en el flujo de audio cuando este flujo debe superar los routers y firewalls. SIP suele necesitar un servidor STUN (*Session Traversal Utilities for NAT*) para estos problemas.
- Estandarización y uso
SIP es un protocolo estandarizado por la IETF hace bastante tiempo y que es ampliamente implementado por todos los fabricantes de equipos y *software*. IAX está aun siendo estandarizado y es por ello que no se encuentra en muchos dispositivos existentes en el mercado.
- Utilización de puertos
IAX utiliza un solo puerto (4569) para mandar la información de señalización y los datos de todas sus llamadas. Para ello utiliza un mecanismo de multiplexión o "*trunking*". SIP, sin embargo utiliza un puerto (5060) para señalización y 2 puertos RTP por cada conexión de audio (como mínimo 3 puertos). Por ejemplo para 100 llamadas simultaneas con SIP se usarían 200 puertos (RTP) más el puerto 5060 de señalización. IAX utilizaría sólo un puerto para todo (4569)
- Flujo de audio al utilizar un servidor

En SIP, si utilizamos un servidor, la señalización de control pasa siempre por el servidor pero la información de audio (flujo RTP) puede viajar extremo a extremo sin tener que pasar necesariamente por el servidor SIP. En IAX al viajar la señalización y los datos de forma conjunta todo el tráfico de audio debe pasar obligatoriamente por el servidor IAX. Esto produce un aumento en el uso del ancho de banda que deben soportar los servidores IAX sobretodo cuando hay muchas llamadas simultáneas.

- Otras funcionalidades

IAX es un protocolo pensado para VoIP y transmisión de video y presenta funcionalidades interesantes como la posibilidad de enviar o recibir planes de marcado (*dialplans*) que resultan muy interesantes al usarse conjuntamente con servidores Asterisk. SIP es un protocolo de propósito general y podría transmitir sin dificultad cualquier información y no sólo audio o video.

Los protocolos que se han descrito son los encargados de enviar la señalización por la red. Para el envío de la voz se usa otro protocolo llamado RTP (*Real-time Transport Protocol*). Es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo. Está desarrollado por el grupo de trabajo de transporte de Audio y Video del IETF, y fue publicado por primera vez como estándar en 1996 como la RFC 1889, y actualizado posteriormente en 2003 en la RFC 3550, que constituye el estándar de Internet STD 64.

Inicialmente se publicó como protocolo *multicast* (envío de la información en múltiples redes a múltiples destinos simultáneamente), aunque se ha usado en varias aplicaciones *unicast* (envío de información desde un único emisor a un único receptor). Se usa frecuentemente en sistemas de *streaming* (distribución de tráfico multimedia por la red de datos), junto a RTSP (*Real Time Streaming Protocol*), videoconferencia y sistemas de llamadas de audio (en conjunción con H.323 o SIP). Representa también la base de la industria de VoIP.

La RFC 1890, obsoleta por la RFC 3551 (STD 65), define un perfil para conferencias de audio y vídeo con control mínimo. La RFC 3711, por otro lado, define SRTP (*Secure Real-time Transport Protocol*), una extensión del perfil de RTP para conferencias de audio y vídeo que puede usarse opcionalmente para proporcionar confidencialidad, autenticación de mensajes y protección de reenvío para flujos de audio y vídeo.

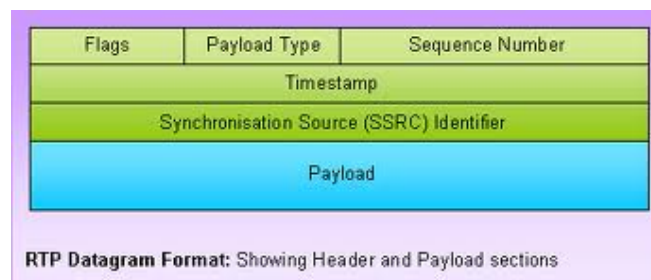


Ilustración 1 - Ejemplo de datagrama RTP

En el campo de datos del paquete RTP es donde se encapsulan los datos de voz de una llamada VoIP codificada con alguno de los codecs que veremos en el presente TFC.

RTP va de la mano de RTCP (*RTP Control Protocol*) y se sitúa sobre UDP en el modelo OSI. Es un protocolo de comunicación que proporciona información de control que está asociado con un flujo de datos para una aplicación multimedia (flujo RTP). Trabaja junto con RTP en el transporte y empaquetado de datos multimedia, pero no transporta ningún dato por sí mismo. Se usa habitualmente para transmitir paquetes de control a los participantes de una sesión multimedia de *streaming*.

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* (efecto de pérdida de voz producido porque algunos paquetes llegan demasiado pronto o tarde y no valen para reconstruir el flujo de voz) 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 (*Quality of Service*). RTCP por sí mismo no ofrece ninguna clase de cifrado de flujo o de autenticación. Para tales propósitos se puede usar SRTCP.

2.2.- ELEMENTOS DE VoIP

Los elementos fundamentales para poder tener una comunicación de VoIP son los siguientes:

- **Cientes:** Son los terminales donde se origina o es destino una llamada. Éstos pueden ser teléfonos al uso tal y como se conocen o un *softphone* (*software* de ordenador que permite usar el mismo como si fuera un teléfono).
- **Servidor:** Es la centralita PBX a la que se conecta el cliente de VoIP y permite realizar llamadas con otros clientes de nuestra red o de otras redes.
- **Gateways:** Son dispositivos que permiten mezclar el mundo de la telefonía tradicional con la tecnología de VoIP. Estos aparatos permiten convertir teléfonos analógicos en terminales VoIP o conectar la red de VoIP a la red telefónica convencional.

La infraestructura más sencilla consistiría en tener una única centralita y dos terminales IP (aunque también se podría establecer entre dos clientes únicamente). A partir de aquí se puede complicar todo lo que se quiera. Además, no es necesario que la centralita esté en la misma red en la que tenemos los teléfonos (ya sean IP o analógicos a través de *gateways*). Esto es muy interesante, ya que se puede pensar en tener un servicio de telefonía ip en la nube (*cloud*).

Actualmente los servicios en la nube se están poniendo muy de moda ya que son más baratos para las empresas que montar las infraestructuras equivalentes en sus instalaciones para poder tener el mismo servicio. Con esta premisa y el poder dar más servicio a más usuarios con wifi en una zona, hacen pensar que pueda ser un buen negocio el de convertirse en operador de VoIP en el *cloud*.

2.3.- DESCRIPCIÓN DE LOS CODECS

Los codecs (acrónimo de codificador-decodificador), son los encargados de convertir la voz analógica en una señal digital que se pueda enviar por una red de datos. Dependiendo del códec elegido se usará un mayor ancho de banda y se tendrá una mejor o peor calidad de audio.

Un ejemplo de codecs usados para esta función es el de la siguiente tabla²:

Nombre	Estandarizado	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones	MOS (Mean Opinion Score)
G.711*	ITU-T	Pulse code modulation (PCM)	64	8	Muestreada	Tiene dos versiones u-law (US, Japan) y a-law (Europa) para muestrear la señal	4.1

² <http://www.voipforo.com/codec/codecs.php>

G.711.1 *	ITU-T	Pulse code modulation (PCM)	80-96Kbps	8	Muestreada	Mejora del códec G.711 para abarcar la banda de 50 Hz a 7 Khz. Mas info	
G.721	ITU-T	Adaptive differential pulse code modulation (ADPCM)	32	8	Muestreada	Obsoleta. Se ha transformado en la G.726.	
G.722	ITU-T	7 kHz audio-coding within 64 kbit/s	64	16	Muestreada	Divide los 16 Khz en dos bandas cada una usando ADPCM	
G.722.1	ITU-T	Codificación a 24 y 32 kbit/s para sistemas sin manos con baja pérdida de paquetes	24/32	16	20		
G.722.2 AMR-WB	ITU-T	Adaptive Multi-Rate Wideband Codec (AMR-WB)	23.85/ 23.05/ 19.85/ 18.25/ 15.85/ 14.25/ 12.65/ 8.85/ 6.6	16	20	Se usa principalmente para compresión de voz en tecnología móvil de tercera generación. Mas info	
G.723	ITU-T	Extensión de la norma G.721 a 24 y 40 kbit/s para aplicaciones en circuitos digitales.	24/40	8	Muestreada	Obsoleta por G.726. Es totalmente diferente de G.723.1.	
G.723.1	ITU-T	Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s	5.6/6.3	8	30	Parte de H.324 video conferencing. Codifica la señal usando linear predictive analysis-by-synthesis coding. Para el codificador de high rate utiliza Multipulse Maximum Likelihood Quantization (MP-MLQ) y para el de low-rate usa Algebraic-Code-Excited Linear-Prediction (ACELP).	3.8-3.9
G.726	ITU-T	40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)	16/24/32/40	8	Muestreada	ADPCM; reemplaza a G.721 y G.723.	3.85
G.727	ITU-T	5-, 4-, 3- and 2-bit/sample embedded adaptive differential pulse code modulation (ADPCM)	var.		Muestreada	ADPCM. Relacionada con G.726.	
G.728	ITU-T	Coding of speech at 16 kbit/s using low-delay code excited linear prediction	16	8	2.5	CELP.	3.61
G.729 **	ITU-T	Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)	8	8	10	Bajo retardo (15 ms)	3.92
G.729.1	ITU-T	Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)	8/12/14/16/ 18/20/22/24/ 26/28/30/32	8	10	Ancho de banda desde 50Hz a 7 Khz Mas info	
GSM 06.10	ETSI	RegularPulse Excitation LongTerm Predictor (RPE-LTP)	13	8	22.5	Usado por la tecnología celular GSM	
LPC10	Gobierno de USA	Linear-predictive codec	2.4	8	22.5	10 coeficientes. La voz suena un poco "robótica"	
Speex			8, 16, 32	2.15- 24.6 (NB) 4-44.2 (WB)	30 (NB) 34 (WB)		
iLBC			8	13.3	30		
DoD CELP	American Department of Defense (DoD) Gobierno de USA		4.8		30		

EVRC	3GPP2	Enhanced Variable Rate CODEC	9.6/4.8/1.2	8	20	Se usa en redes CDMA	
DVI	Interactive Multimedia Association (IMA)	DVI4 uses an adaptive delta pulse code modulation (ADPCM)	32	Variable	Muestreada		
L16		Uncompressed audio data samples	128	Variable	Muestreada		
SILK	Skype	Uncompressed audio data samples	De 6 a 40 kbit/s	Variable	20	El códec Harmony está basado en SILK	

Tabla 1 - Resumen de codecs de voz

* El códec g711 tiene dos versiones conocidas como alaw (usado en Europa) y ulaw (usado en USA y Japón). U-law se corresponde con el estándar T1 usado en Estados Unidos y A-law con el estándar E1 usado en el resto del mundo. La diferencia es el método que se utiliza para muestrear la señal. La señal no se muestrea de forma lineal sino de forma logarítmica. A-law tiene un mayor rango. Para mas información de las diferencias ver [G.711 Ley A vs Ley u](#)

**existen varias versiones del códec g729 que es interesante explicar por su extendido uso

G729: es el códec original

G729A o anexo A: es una simplificación de G729 y es compatible con G729. Es menos complejo pero tiene algo menos de calidad.

G729B o anexo B: Es G729 pero con supresión de silencios y no es compatible con las anteriores.

G729AB: Es g729A con supresión de silencios y sería compatible solo con G729B.

Aparte de esto G729 (todas las versiones) en general tienen un bit rate de 8Kbps pero existen versiones de 6.4 kbps (anexo D) y 11.4 Kbps (anexo E).

Los más comunes, y usados para la simulación posterior, son los codecs G.711, G.723.1 y G.729. Se verán un poco más en detalle estos codecs para entender su funcionamiento. El hecho de que estos protocolos sean los más usados es porque G.711 es un códec que es igual que las implementaciones que se siguen en la telefonía tradicional. La calidad de voz en él es muy buena y además es muy apto para la transmisión de los tonos (llamados también DTMF) y para el envío de fax usando el protocolo T.38.

Por su parte, G.723.1 y G.729 son protocolos muy usados debido a que la relación de ancho de banda consumido y la calidad de audio que tienen es bastante buena. Esto ha hecho que se popularicen mucho a pesar de ser codecs propietarios y de los que hay que pagar patente para poder usarlos. Pero el menor consumo de ancho de banda los hace muy aptos para usos en redes con poco ancho de banda o para conseguir tener más comunicaciones en la misma red que si se usara el códec G.711.

Además, el precio que se puede ahorrar en comunicaciones compensa el tener que pagar las licencias debidas a las patentes de estos codecs. Como se verá luego en la simulación, el número de llamadas que se puede llegar a alcanzar con el mismo ancho de banda es mucho mayor para estos codecs que para el G.711.

2.4.- CÓDEC G.711

G.711 es un códec de audio definido por el ITU-T en el año 1972. También es conocido por el nombre de PCM (*Pulse Code Modulation*). Es un estándar usado en telefonía principalmente. Este códec permite el paso de la voz comprendida entre los 300Hz y los 3400Hz, tomando 8000 muestras por segundo, lo que permite tener una tasa de transferencia de 64kbit/s. Tiene dos variantes ligeramente diferentes: la Ley- μ -usada principalmente en Norte América- y la Ley-A -usada en casi todos los países fuera de Norte América-. La mayor diferencia entre estos dos tipos es que la Ley- μ codifica una muestra de 14 bits con signo en una muestra de 8bits, mientras que la Ley-A codifica una muestra de 13 bits con signo en una de 8 bits.

Características:

- Frecuencia de muestreo de 8KHz.
- Tasa de transferencia de 64kbit/s.
- El retardo típico del algoritmo es de 0,125 ms.

- G711 es un codificador de voz de forma de onda.
- El apéndice I de G711 define un algoritmo de PLC (*Packet Loss Concealment*) para ayudar a ocultar las pérdidas de transmisión en una red de paquetes.
- El apéndice II de G711 define un algoritmo DTX (*Discontinuous Transmission*) el cual usa VAD (*Voice Activity Detection*) y CNG (*Confort Noise Generation*) para reducir el ancho de banda durante los periodos de silencio.

Este códec es el que mejor calidad de audio da, ya que es la misma codificación que se usa en la telefonía tradicional. Aunque usa un mayor ancho de banda que los otros codecs, permite también el envío de faxes y de tonos DTMF, cosa que los otros dos codecs no son aptos para este envío de información.

Existe una variante de este códec llamada G.711.1 que mejora aún más la calidad de audio pero aumentando el ancho de banda consumido. Esto lo consigue muestreando más frecuencias de las mencionadas anteriormente.

Además, este códec es gratuito, ya que las patentes caducaron hace muchos años, lo cual hace que esté implementado en todos los dispositivos que se necesiten para una infraestructura de VoIP. Incluso los *softphone* gratuitos lo traen integrado.

2.5.- CÓDEC G.723.1

G.723.1 es un códec de audio, aprobado en mayo de 2006, que comprime la voz en tramas de 30 ms. Además, hay un preanálisis de 7,5 ms de duración, lo que hace que el retardo final sea de 37,5 ms. El uso más frecuente de este códec es en VoIP, ya que sus necesidades bajas de ancho de banda (5,3 ó 6,3 kbit/s) lo hacen ideal para este fin.

Características:

- Frecuencia de muestreo de 8Khz/16-bit (240 muestras por cada trama de 30 ms).
- Tasa de transferencia fija (5,3 kbit/s con 20 bytes por trama de 30 ms; 6,3 kbit/s con 24 bytes por trama de 30 ms).
- Tamaño fijo de trama para cada tasa de transferencia.
- El retardo algorítmico por trama es de 37,5 ms, con el retardo de preanálisis de 7,5 ms.

Este códec es más sensible a los retardos que el G.711, ya que los algoritmos de compresión son más complejos. Esto provoca que si no se tiene capacidad de procesamiento suficiente para estos cálculos, se produzca una degradación en la calidad de la comunicación.

Aún así, a pesar de este pequeño inconveniente, el poco uso que hace de ancho de banda, lo convierten en un gran candidato para comunicaciones en redes de largo alcance, donde el ancho de banda disponible no es muy grande.

En el caso de este códec, al no ser gratuito, no se encuentra implementado en todos los dispositivos que se usen en una infraestructura de VoIP. En los *softphone* gratuitos se tiene que pagar una licencia a parte si se quiere hacer uso del códec.

2.5.1- DESCRIPCIÓN DE LA PARTE CODIFICADORA

Este códec está diseñado para el funcionamiento con una señal digital obtenida filtrando primero la entrada analógica con la anchura de banda de telefonía (Rec. ITU-T G.712), muestreándola luego a 8000 Hz y convirtiéndola a señal PCM lineal de 16 bit para su entrada en el codificador. Habrá que convertir la

salida del decodificador a señal analógica mediante medios similares. Otras características entrada/salida, como las especificadas en la Rec. ITU-T G.711 para los datos PCM a 64 kbit/s, se convertirán a PCM lineal a 16 bit antes de la codificación o de PCM lineal a 16 bit al formato apropiado después de la decodificación. En esta recomendación se define el tren de bits que va del codificador hasta el decodificador.

El códec se basa en los principios de la codificación de predicción lineal y análisis por síntesis, y trata de hacer mínima una señal de error ponderada perceptualmente. El codificador funciona con bloques (tramas) de 240 muestras cada uno. Ello equivale a 30 ms a una velocidad de muestreo de 8 kHz. Cada bloque se pasa primero por un filtro paso alto para suprimir la componente continua, y luego se divide en cuatro subtramas de 60 muestras cada una. Para cada subtrama, se calcula un filtro de códec de predicción lineal (LPC, *Linear Prediction Coder*) de décimo orden utilizando la señal de entrada no procesada. El filtro LPC para la última subtrama se cuantifica con un cuantificador vectorial de división predictiva (PSVQ, *Predictive Split Vector Quantizer*). Los coeficientes LPC no cuantificados se utilizan para construir el filtro de ponderación perceptual de corto plazo, que se utiliza para filtrar toda la trama y obtener la señal de voz ponderada perceptualmente.

Para cada dos subtramas (120 muestras), se calcula el periodo de tono en bucle abierto, L_{OL} , mediante la señal vocal ponderada. Esta estimación del tono se realiza con bloques de 120 muestras. El periodo de tono se busca en la gama de 18 a 142 muestras. A partir de ese punto, la voz se procesa a 60 muestras por subtramas.

Utilizando el periodo de tono estimado calculado anteriormente, se construye un filtro de conformación de ruido armónico. La combinación del filtro de síntesis LPC, el filtro de ponderación perceptual formante y el filtro de conformación del ruido armónico se utiliza para crear una respuesta de impulso. La respuesta del impulso se utiliza para los cálculos posteriores. Con la estimación del periodo de tono, L_{OL} , y la respuesta de impulso, se calcula un predictor de tono en bucle cerrado. Se utiliza un predictor de tono de quinto orden. El periodo de tono se calcula como un valor diferencial pequeño respecto de la estimación de tono en bucle abierto. La contribución del predictor de tono se resta del vector objetivo inicial. Tanto el periodo de tono como el valor diferencial se transmiten al decodificador.

Por último, se aproxima la componente no periódica de la excitación. Para la velocidad alta, se utiliza la excitación del tipo cuantificación multiimpulso de máxima verosimilitud (MP-MLQ, *Multipulse Maximum Likelihood Quantization*), y para la velocidad baja, una predicción lineal con excitación por tabla de códigos algebraicos (ACELP, *Algebraic Code Excited Linear Prediction*).

audio. Los dos anexos más comunes son el A y el B. El primero de ellos utiliza una menor complejidad a la hora de realizar la codificación, pero esto conlleva un empeoramiento marginal en la calidad de la conversación. Este anexo es compatible con el códec original. Por otro lado, el anexo B se encarga de realizar una compresión del silencio para optimizar más el uso de ancho de banda.

Características:

- Frecuencia de muestreo de 8KHz/16-bit (80 muestras por cada trama de 10 ms).
- Tasa de transferencia fija (8 kbit/s en cada trama de 10 ms).
- Tamaño fijo de trama (10 bytes por cada trama de 10 ms).
- El retardo del algoritmo es de 15 ms, con un retardo de preanálisis de 5 ms.

Este códec también es sensible a los retardos como le ocurre al G.723.1, pero en menor medida. La complejidad de los algoritmos de compresión es menor que en el otro códec, pero si se tienen retardos debidos a los cálculos que hay que hacer, se notará una sensación de mala calidad de audio.

También es un códec con licenciamiento, con lo que si se usa un *softphone* gratuito, se debe pagar una licencia adicional para poder usar este códec. Por otro lado, casi todos los dispositivos *hardware* de VoIP ya incluyen la licencia de G.729 para poder usarlo, lo cual hace que sea un códec muy utilizado.

Además, debido al poco uso que hace de ancho de banda, y que la calidad de audio que tiene es bastante buena, lo convierten en un códec muy utilizado para comunicaciones de VoIP en redes con recursos más limitados, ya que se pueden realizar más comunicaciones que con G.711 y manteniendo una buena calidad de audio.

2.6.1- DESCRIPCIÓN DE LA PARTE CODIFICADORA

El principio de codificación se muestra en la Ilustración 3. La señal de entrada se pasa por un filtro paso alto y se amplifica en el bloque de preprocesamiento. La señal de preprocesado sirve como señal de entrada para todos los análisis posteriores. El análisis LP (*Linear Prediction*) se realiza una vez por cada trama de 10 ms para calcular los coeficientes del filtro LP. Estos coeficientes se convierten en pares de espectro de línea (LSPs -*Line Spectrum Pairs*) y cuantificados usando un cuantificador vectorial predictivo de dos etapas (VQ -*Vector Quantization*) con 18 bits.

La señal de excitación se escoge mediante el uso de un procedimiento de búsqueda de análisis por síntesis en la que se reduce al mínimo el error entre la voz original y la reconstruida de acuerdo a una medida de la distorsión ponderada perceptualmente. Esto se hace mediante el filtrado de la señal de error con un filtro de ponderación perceptual, cuyos coeficientes se derivan de filtro LP no cuantificados. La cantidad de ponderación perceptual se hace adaptativa para mejorar el rendimiento de las señales de entrada con una respuesta en frecuencia plana.

Los parámetros de excitación (tanto las tablas de códigos fijos como adaptativos) se determinan por cada subtrama de 5 ms (40 muestras). Los coeficientes cuantificados y no cuantificados del filtro LP se utilizan para la segunda subtrama, mientras que en la primera subtrama se interpolan los coeficientes del filtro LP (tanto cuantificados como no cuantificados). El retardo de tono en bucle abierto se calcula una vez para cada trama de 10 ms basado en la señal de voz ponderada perceptualmente.

A continuación, las siguientes operaciones se repiten para cada subtrama. La señal objetivo $x(n)$ se calcula filtrando la LP residual a través del filtro de síntesis ponderado. Los estados iniciales de estos filtros se actualizan filtrando el error entre la LP residual y la excitación. Esto es equivalente a la aproximación común de restar la respuesta de entrada cero del filtro de síntesis ponderado de la señal de voz ponderada. Se calcula la respuesta impulsional $h(n)$ del filtro de síntesis ponderado.

A continuación se realiza el análisis de tono en bucle cerrado (para encontrar el retardo de las tablas de códigos adaptativos y la ganancia), utilizando la señal objetivo $x(n)$ y la respuesta impulsional $h(n)$, mediante la búsqueda en todo el valor del retardo de tono en bucle abierto. Se utiliza un retardo de tono fraccionario con una resolución de $1/3$. El retardo de tono se codifica con 8 bits en la primera subtrama y diferencialmente con 5 bits en el segunda.

La señal objetivo $x(n)$ se actualiza restando la contribución (filtrada) de las tablas de códigos adaptativos, y esta nueva señal, $x'(n)$, se utiliza en la búsqueda de las tablas de códigos fijos para encontrar la excitación óptima. Se utilizan unas tablas de códigos algebraicos de 17 bits para la excitación de las tablas de códigos fijos. Las ganancias de las contribuciones de las tablas de códigos adaptativos y fijos son cuantificadas vectorialmente con 7 bits (con predicción MA aplicada a la ganancia de las tablas de códigos fijos). Finalmente, las memorias de los filtros se actualizan utilizando la señal de excitación determinada.

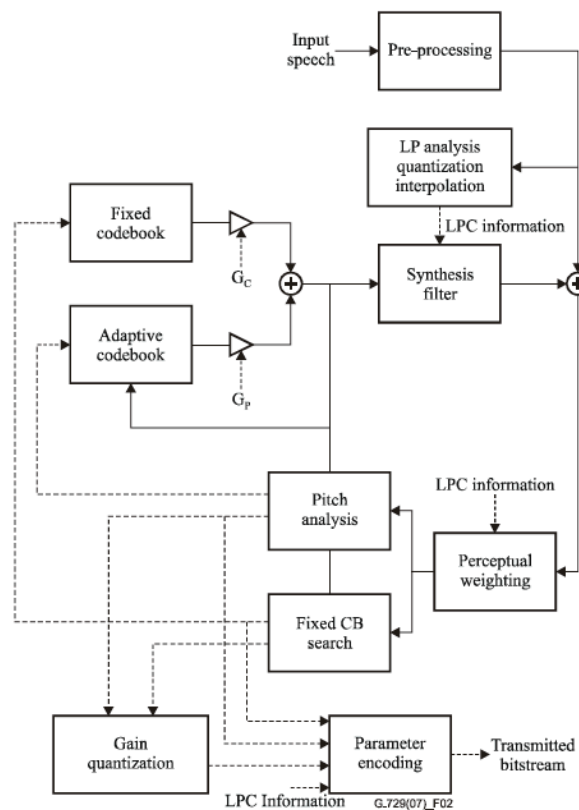


Ilustración 4 - Diagrama de bloques del codificador G.729

2.6.1- DESCRIPCIÓN DE LA PARTE DECODIFICADORA

El principio del decodificador se muestra en la Ilustración 4. En primer lugar, los índices de los parámetros se extraen del flujo de bits recibido. Estos índices se descodifican para obtener los parámetros del codificador que corresponden a una trama de voz de 10 ms. Estos parámetros son los coeficientes LSP, los dos retardos de tono fraccionarios, los dos vectores de tablas de códigos fijos, y los dos conjuntos de ganancias de las tablas de códigos adaptativos y fijos. Los coeficientes LSP se interpolan y se convierten en los coeficientes del filtro LP para cada subtrama. Entonces, para cada subtrama de 5 ms se realizan los siguientes pasos:

- la excitación se construye añadiendo los vectores de las tablas de códigos adaptativos y fijos escalados por sus respectivas ganancias
- la voz se reconstruye mediante el filtrado de la excitación a través del filtro de síntesis LP
- la señal de voz reconstruida se pasa a través de una etapa de post-procesamiento, que incluye un postfiltro adaptativo basado en los filtros de síntesis de larga duración y de corta duración, seguido por un filtro paso alto y una operación de escalado

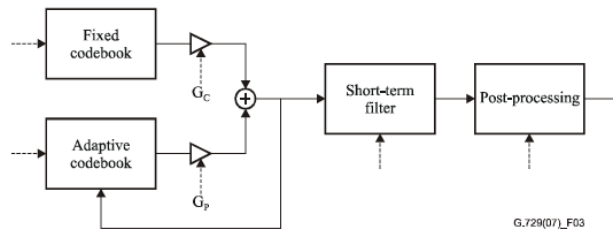


Ilustración 5 - Diagrama de bloques del decodificador de G.729

2.7.- RESUMEN

Una vez visto cómo funcionan los codecs que se van a usar en la simulación, conviene ver un pequeño resumen de las ventajas y desventajas que tiene cada uno. Como la simulación que se va a realizar se basa fundamentalmente en el uso del ancho de banda disponible, se pone una tabla de los codecs viendo el uso de ancho de banda que tienen:

Codec Information				Bandwidth Calculations					
Codec & Bit Rate (Kbps)	Codec Sample Size (Bytes)	Codec Sample Interval (ms)	Mean Opinion Score (MOS)	Voice Payload Size (Bytes)	Voice Payload Size (ms)	Packets Per Second (PPS)	Bandwidth MP or FRF.12 (Kbps)	Bandwidth w/cRTP MP or FRF.12 (Kbps)	Bandwidth Ethernet (Kbps)
G.711 (64 Kbps)	80 Bytes	10 ms	4.1	160 Bytes	20 ms	50	82.8 Kbps	67.6 Kbps	87.2 Kbps
G.729 (8 Kbps)	10 Bytes	10 ms	3.92	20 Bytes	20 ms	50	26.8 Kbps	11.6 Kbps	31.2 Kbps
G.723.1 (6.3 Kbps)	24 Bytes	30 ms	3.9	24 Bytes	30 ms	33.3	18.9 Kbps	8.8 Kbps	21.9 Kbps
G.723.1 (5.3 Kbps)	20 Bytes	30 ms	3.8	20 Bytes	30 ms	33.3	17.9 Kbps	7.7 Kbps	20.8 Kbps

Tabla 2 - Resumen de características de ancho de banda de los codecs

En esta tabla se muestra claramente el ahorro de ancho de banda que se tiene con los codecs G.729 y G.723.1 comparado con el G.711. Este hecho es muy importante, ya que va a permitir poder cursar muchas más llamadas con estos codecs usando los mismos recursos. El problema que tienen frente al G.711 es que son más sensibles a los retardos y no se pueden transmitir los códigos DTMF (códigos de marcado) y los faxes usando el códec en sí mismo. Por ello hay que buscar otras técnicas para poder hacer uso de ellos.

En mi propia experiencia, el códec G.729 es el más usado de todos. Ello se debe a que tiene una muy buena calidad en las llamadas, y si se realiza un buen dimensionamiento de la red y de la configuración de las calidades de servicio, se pueden paliar los problemas que tiene el retardo en el envío de los paquetes con información. El ahorro de ancho de banda, unido a la calidad de las llamadas, son lo que convierten a este códec en la mejor elección para las llamadas de VoIP.

Además, para el envío de los códigos DTMF se pueden usar otras técnicas. A continuación se verán las técnicas usadas para poder enviar este tipo de tonos:

- DTMF: SIP *info*

El método de SIP *info* (RFC 2976) puede ser usado por los elementos de la red para transmitir tonos DTMF fuera de la banda (*out-of-band*) como eventos de teléfono (*telephone-events*) de manera fiable e independiente del tráfico de audio.

- DTMF: "*in band*"

Como su nombre indica, en este método los tonos DTMF se envían en el flujo de datos de la comunicación o códec de RTP. Este es un método muy rudimentario, los tonos DTMF se envían como una voz normal y no tenemos ningún control sobre los mismos, no los podemos ver o decodificar usando un analizador de protocolos.

Los codecs de voz con compresión como G.729, G.723.1, etc., que utilizan mucho menos ancho de banda limitan también el ancho de banda de las señales DTMF y estas no se transfieren adecuadamente, con lo cual el que las recibe no es capaz de decodificarlas. Por ello se creó el método del RFC 2833

- DTMF: RFC 2833

El RFC 2833 define básicamente una nueva carga que ha sido creado especialmente para el envío de los tonos DTMF, de esta manera los tonos DTMF no dependen de un códec que fue diseñado para la voz, pues tiene su propio códec optimizado soportado por la RFC 2833.

El RFC 2833 determina cómo usar la carga útil del paquete de RTP para los tonos (DTMF) y señales de telefonía.

El RFC 2833 está siendo reemplazado por el RFC 4733, el cual es más complejo y adquiere nuevas prestaciones.

3.- DESCRIPCIÓN DE LOS PROTOCOLOS DE WIRELESS 802.11

El estándar 802.11 define una serie de protocolos para las capas física (PHY - *physical layer*) y de enlace (MAC - *Media Access Control*) del modelo OSI para poder conectar dispositivos a una red *wireless*. La primera versión del protocolo se publicó en 1997 y ha tenido varias correcciones. Las que más se han impuesto son las normas 802.11a, 802.11b, 802.11e, 802.11g y 802.11n (esta última más en la actualidad). Las normas a, b, g y n definen las velocidades de transmisión que puede manejar el protocolo. La norma e define la Calidad de Servicio (QoS por sus siglas en inglés - *Quality of Service*) para poder garantizar la transmisión de tráfico multimedia a través de la red *wireless*. En la norma original se definieron tasas de transferencia de 1 y 2 Mbit/s.

El uso de la tecnología *wireless* ha supuesto una revolución en las comunicaciones, ya que han permitido dar movilidad a las personas y mantenerlas conectadas a la red y sus servicios. Actualmente ya no se concibe un dispositivo que no disponga de conexión *wireless*, ya que casi nadie lo utilizaría. Esto ha llevado a que la tecnología WiFi se desarrolle mucho y esté en continuo crecimiento, ofreciendo a los usuarios mayores velocidades de transmisión, llegando actualmente a alcanzar casi las mismas velocidades que las conexiones con cable.

Aunque no es objetivo de este trabajo, se ha definido la norma ac que ya casi alcanza la velocidad de 1Gb/s, que actualmente es una velocidad muy normal en las conexiones cableadas. Con las siguientes evoluciones, se conseguirá que ya no se necesite tener un equipo conectado con cable para conseguir un rendimiento similar.

A modo informativo se incluye la siguiente tabla con un resumen de los protocolos del 802.11³:

802.11 network PHY standards										
802.11 protocol	Release ^[19]	Freq. (GHz)	Bandwidth (MHz)	Data rate per stream (Mbit/s) ^[20]	Allowable MIMO streams	Modulation	Approximate indoor range		Approximate outdoor range	
							(m)	(ft)	(m)	(ft)
—	Jun 1997	2.4	22	1, 2	N/A	DSSS, FHSS	20	66	100	330
a	Sep 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	N/A	OFDM	35	115	120	390
		3.7 ^[21]					—	—	5,000	16,000 ^[21]
b	Sep 1999	2.4	22	1, 2, 5.5, 11	N/A	DSSS	35	115	140	460
g	Jun 2003	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	N/A	OFDM, DSSS	38	125	140	460
n	Oct 2009	2.4/5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2 ^[22]	4	OFDM	70	230	250	820 ^[21]
			40	15, 30, 45, 60, 90, 120, 135, 150 ^[21]			70	230	250	820 ^[21]
ac	Dec 2013	5	20	up to 87.6 ^[22]	8					
			40	up to 200 ^[22]						
			80	up to 433.3 ^[22]						
			160	up to 866.7 ^[22]						
ad	Dec 2012	60	2,160	up to 6912 (6.75Gb/s) ^[23]	1	Single Carrier, OFDM, Low-Power Single Carrier				
ah	Est. 2016 ^[19]	900MHz								
aj	Est. 2016 ^[19]	45/60								
ax	TBD	2.4/5				OFDM				

Tabla 3 - Resumen de familia de protocolos 802.11

³ http://en.wikipedia.org/wiki/IEEE_802.11n-2009

3.1.- ESTÁNDAR 802.11A

Esta norma fue definida en 1999 y se basa en una multiplexación OFDM (*Orthogonal Frequency Division Multiplexing*) para conseguir unas velocidades comprendidas entre 1,5 y 54 Mbit/s. La banda frecuencial en la que trabaja es la de 5GHz. Tiene asignados 12 canales sin solapamiento con un ancho de banda de 20MHz. El hecho de trabajar en los 5GHz le hace incompatible con las normas b y g, salvo que se disponga de dispositivos que sepan manejar todos los estándares.

El hecho de usar los 5GHz es porque en EEUU es una banda frecuencial de uso libre, mientras que en Europa es de uso restringido. Es por ello que el estándar a no ha tenido mucho éxito en Europa mientras que el b y el g sí. A pesar de esto, el uso de la banda de los 5GHz evita las interferencias que se producen en la banda de 2,4GHz (hornos microondas, teléfonos inalámbricos...), pero se requiere más potencia de emisión para intentar tener tanto alcance como en la de 2,4GHz.

A pesar de que se puede llegar a transmitir a 54Mbit/s, la tasa normal de transferencia se sitúa en los 20Mbit/s. Las ventajas que puede parecer tener el trabajar en una frecuencia con menor interferencia no fue suficiente para aventajar a los estándares b y g, ya que los dispositivos que los implementaban eran más baratos. Este hecho hizo que se propagasen más los dispositivos que implementaban los estándares b y g frente a los que implementaban el a.

3.2.- ESTÁNDAR 802.11B

Esta norma fue definida también en 1999 igual que la 802.11a y usa CSMA/CA(*Carrier Sense Multiple Access with Collision Avoidance*) como método de acceso al medio al igual que la norma original. Puede alcanzar velocidades de transferencia de hasta 11Mbit/s, aunque se ha demostrado que con el uso del protocolo TCP puede llegar hasta los 5,9Mbit/s y si se usa el protocolo UDP hasta los 7,1Mbit/s.

Gracias a este aumento de velocidad y al precio reducido que tenían los dispositivos que implementaban este estándar, el crecimiento fue muy rápido al igual que la implantación en casi todos los países del mundo.

Este estándar trabaja en la banda de frecuencias de los 2,4GHz, igual que la original, lo que provoca que sufra interferencias producidas por otros dispositivos que también funcionan en la misma banda frecuencial. Además, el ancho de banda de cada canal es de 22MHz, y usa hasta un total de 14 canales dependiendo del país. Esto provoca que se produzca solapamiento de canales. En la siguiente tabla se ve este hecho⁴:

Channel	Center Frequency	Frequency delta	Channel Width	Overlaps Channels
1	2.412 GHz	5 MHz	2.401–2.423 GHz	2-5
2	2.417 GHz	5 MHz	2.406–2.428 GHz	1,3-6
3	2.422 GHz	5 MHz	2.411–2.433 GHz	1-2,4-7
4	2.427 GHz	5 MHz	2.416–2.438 GHz	1-3,5-8
5	2.432 GHz	5 MHz	2.421–2.443 GHz	1-4,6-9
6	2.437 GHz	5 MHz	2.426–2.448 GHz	2-5,7-10
7	2.442 GHz	5 MHz	2.431–2.453 GHz	3-6,8-11
8	2.447 GHz	5 MHz	2.436–2.458 GHz	4-7,9-12
9	2.452 GHz	5 MHz	2.441–2.463 GHz	5-8,10-13
10	2.457 GHz	5 MHz	2.446–2.468 GHz	6-9,11-13
11	2.462 GHz	5 MHz	2.451–2.473 GHz	7-10,12-13
12	2.467 GHz	5 MHz	2.456–2.478 GHz	8-11,13-14

⁴ http://en.wikipedia.org/wiki/IEEE_802.11b-1999

13	2.472 GHz	5 MHz	2.461–2.483 GHz	9-12, 14
14	2.484 GHz	12 MHz	2.473–2.495 GHz	12-13

Tabla 4 - Canales usados por 802.11b

El canal 14 sólo está permitido en Japón. Los canales 12 y 13 están permitidos en casi todas las partes del mundo excepto en EEUU, donde sólo es legal el uso de los canales del 1 al 11. En esta tabla además se ve el efecto del solapamiento de canales. De todas formas, este efecto se aprecia mejor en la siguiente figura⁵:

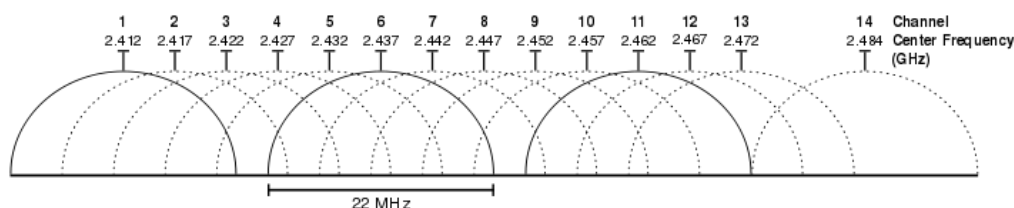


Ilustración 6 - Solapamiento de canales en la banda de 2,4 GHz

Aquí se aprecia cómo se solapan los canales tal y como se indica en la tabla anterior. Se recomienda el uso de los canales 1, 6 y 11 con separación de 25MHz como canales sin solapamiento en EEUU y los canales 1, 5, 9 y 13 con separación de 20MHz en Europa.

3.3.- ESTÁNDAR 802.11g

En junio de 2003 se ratificó un tercer estándar de modulación que era una evolución del 802.11b. Este nuevo estándar alcanza velocidades de hasta 54Mbit/s, pero en promedio llega hasta los 22Mbit/s, similar al del estándar 802.11a. También utiliza la banda de frecuencias de 2,4GHz, lo que hace que también sea sensible a interferencias como ya se ha comentado anteriormente.

Este estándar es compatible con el 802.11b, lo que hizo que evolucionase muy deprisa ya que se pudieron aprovechar todos los diseños hechos para el estándar anterior. El problema que tiene esta compatibilidad, es que si hay en la misma red dispositivos funcionando con el estándar g y otros con el b, la velocidad de toda la red se ve forzada a ir a la velocidad del protocolo b, con lo que no se nota la mejoría del g. Este hecho se produce debido a que el estándar b no comprende OFDM(*Orthogonal Frequency Division Multiplexing*) que es lo que usa el 802.11g para alcanzar las altas velocidades que usa (igual que hacía el estándar 802.11a).

La red utiliza unos mecanismos entre los puntos de acceso y los propios dispositivos, para informar a la red de si hay algún miembro de la red funcionando con el estándar 802.11b. Para ello utiliza unas tramas con información llamada ERP (*Extended Rate Physical*), las cuales dicen si hay que usar el estándar g o el b para poder adaptarse al tipo de cliente conectado.

Como se ha comentado anteriormente, al ser un estándar compatible con el 802.11b, también hace uso de 14 canales de 22MHz de ancho de banda cada uno, sobre la banda de los 2,4GHz. Al igual que ocurre con 802.11b, en el g también se produce el efecto del solapamiento de los canales⁶:

⁵ http://en.wikipedia.org/wiki/List_of_WLAN_channels

⁶ http://en.wikipedia.org/wiki/IEEE_802.11g-2003

Channel	Center Frequency	Channel Width	Overlaps Channels
1	2.412 GHz	2.401 GHz - 2.423 GHz	2,3,4,5
2	2.417 GHz	2.406 GHz - 2.428 GHz	1,3,4,5,6
3	2.422 GHz	2.411 GHz - 2.433 GHz	1,2,4,5,6,7
4	2.427 GHz	2.416 GHz - 2.438 GHz	1,2,3,5,6,7,8
5	2.432 GHz	2.421 GHz - 2.443 GHz	1,2,3,4,6,7,8,9
6	2.437 GHz	2.426 GHz - 2.448 GHz	2,3,4,5,7,8,9,10
7	2.442 GHz	2.431 GHz - 2.453 GHz	3,4,5,6,8,9,10,11
8	2.447 GHz	2.436 GHz - 2.458 GHz	4,5,6,7,9,10,11,12
9	2.452 GHz	2.441 GHz - 2.463 GHz	5,6,7,8,10,11,12,13
10	2.457 GHz	2.446 GHz - 2.468 GHz	6,7,8,9,11,12,13
11	2.462 GHz	2.451 GHz - 2.473 GHz	7,8,9,10,12,13
12	2.467 GHz	2.456 GHz - 2.478 GHz	8,9,10,11,13,14
13	2.472 GHz	2.461 GHz - 2.483 GHz	9,10,11,12,14
14	2.484 GHz	2.473 GHz - 2.495 GHz	12,13

Tabla 5 - Canales usados por 802.11 b/g

Al igual que en el estándar 802.11b, no todos los canales se pueden usar en todos los países.

El estándar g también creció muy deprisa debido a la compatibilidad con el 802.11b, tal y como se ha dicho anteriormente. Durante bastante tiempo los equipos portátiles no incluían dispositivos de *wireless* integrados, pero para poder hacer uso de esta tecnología se usaban unas tarjetas especiales llamadas PCMCIA (*Personal Computer Memory Card International Association*). En ellas se integraban los chips necesarios para poder hacer uso de las redes *wireless*. Este tipo de dispositivos ya no se usan prácticamente nada ya que la inmensa mayoría de dispositivos móviles integran los dispositivos de *wireless*.

3.4.- ESTÁNDAR 802.11N

El estándar 802.11n se comienza a anunciar en enero de 2004. En octubre de 2009 se publica la versión definitiva, aunque algunos fabricantes ya empezaron a integrar las versiones previas del protocolo, con promesas de actualizaciones cuando se publicase la versión definitiva.

802.11n se desarrolla como un protocolo compatible con sus predecesores, tanto el a como el g (siendo compatible con el g también lo es del b). Esto es lo que hizo que se tardase tanto en desarrollar.

Las velocidades que puede desarrollar este estándar van desde los 54Mbit/s hasta los 600Mbit/s. Para poder conseguirlo hace uso de una tecnología llamada MIMO (*Multiple Input - Multiple Output*), la cual consiste en el uso de varias antenas para poder aumentar las velocidades. En el caso de este protocolo se pueden llegar a usar hasta cuatro antenas de manera simultánea. También incluye el uso de canales de 40MHZ en la capa PHY y el uso de agregación de tramas en la capa MAC.

Las bandas de frecuencias en las que trabaja son las de 2,4GHz y 5GHz (por eso es compatible con todos los estándares anteriores). Las capacidades máximas de 802.11n se desarrollan en esta banda de 5GHz cuando se usan canales de 40MHz de ancho de banda y el uso máximo de MIMO con 4 flujos continuos de transmisión. Todas las velocidades se pueden ver en la siguiente tabla⁷:

⁷ http://en.wikipedia.org/wiki/IEEE_802.11n-2009

MCS index	Spatial streams	Modulation type	Coding rate	Data rate (Mbit/s)			
				20 MHz channel		40 MHz channel	
				800 ns GI	400 ns GI	800 ns GI	400 ns GI
0	1	BPSK	1/2	6.50	7.20	13.50	15.00
1	1	QPSK	1/2	13.00	14.40	27.00	30.00
2	1	QPSK	3/4	19.50	21.70	40.50	45.00
3	1	16-QAM	1/2	26.00	28.90	54.00	60.00
4	1	16-QAM	3/4	39.00	43.30	81.00	90.00
5	1	64-QAM	2/3	52.00	57.80	108.00	120.00
6	1	64-QAM	3/4	58.50	65.00	121.50	135.00
7	1	64-QAM	5/6	65.00	72.20	135.00	150.00
8	2	BPSK	1/2	13.00	14.40	27.00	30.00
9	2	QPSK	1/2	26.00	28.90	54.00	60.00
10	2	QPSK	3/4	39.00	43.30	81.00	90.00
11	2	16-QAM	1/2	52.00	57.80	108.00	120.00
12	2	16-QAM	3/4	78.00	86.70	162.00	180.00
13	2	64-QAM	2/3	104.00	115.60	216.00	240.00
14	2	64-QAM	3/4	117.00	130.00	243.00	270.00
15	2	64-QAM	5/6	130.00	144.40	270.00	300.00
16	3	BPSK	1/2	19.50	21.70	40.50	45.00
17	3	QPSK	1/2	39.00	43.30	81.00	90.00
18	3	QPSK	3/4	58.50	65.00	121.50	135.00
19	3	16-QAM	1/2	78.00	86.70	162.00	180.00
20	3	16-QAM	3/4	117.00	130.00	243.00	270.00
21	3	64-QAM	2/3	156.00	173.30	324.00	360.00
22	3	64-QAM	3/4	175.50	195.00	364.50	405.00
23	3	64-QAM	5/6	195.00	216.70	405.00	450.00
24	4	BPSK	1/2	26.00	28.80	54.00	60.00
25	4	QPSK	1/2	52.00	57.60	108.00	120.00
26	4	QPSK	3/4	78.00	86.80	162.00	180.00
27	4	16-QAM	1/2	104.00	115.60	216.00	240.00
28	4	16-QAM	3/4	156.00	173.20	324.00	360.00
29	4	64-QAM	2/3	208.00	231.20	432.00	480.00
30	4	64-QAM	3/4	234.00	260.00	486.00	540.00
31	4	64-QAM	5/6	260.00	288.80	540.00	600.00
32	1	BPSK	1/2	N/A	N/A	6.50	7.20

Tabla 6 - Velocidades de 802.11n según índice MCS

El índice MCS (*Modulation and Coding Scheme*) define unos códigos con los que se interpretan las diferentes tasas de velocidad según los parámetros que se pueden definir en el estándar. Aquí aparece también un parámetro llamado GI (*Guard Interval*), que define el tiempo entre símbolos, y que como se aprecia en la tabla influye en la velocidad que se consigue finalmente.

Como la banda de los 2,4GHZ está muy congestionada, en el estándar n se define la posibilidad de usar el ancho de banda de los 40MHz para poder mejorar las velocidades. Para poder conseguir esto se define el uso de un segundo canal de 20MHz también de ancho de banda tal y como se muestra en la siguiente tabla⁸:

⁸ http://en.wikipedia.org/wiki/IEEE_802.11n-2009

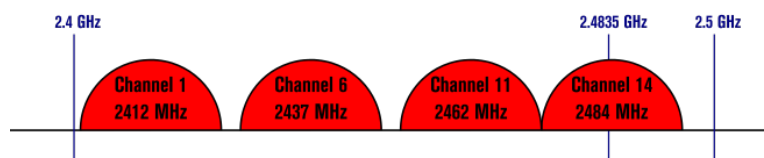
Primary channel	20 MHz	40 MHz above			40 MHz below		
	Blocks	2nd ch.	Center	Blocks	2nd ch.	Center	Blocks
1	1-3	5	3	1-7	Not Available		
2	1-4	6	4	1-8	Not Available		
3	1-5	7	5	1-9	Not Available		
4	2-6	8	6	2-10	Not Available		
5	3-7	9	7	3-11	1	3	1-7
6	4-8	10	8	4-12	2	4	1-8
7	5-9	11	9	5-13	3	5	1-9
8	6-10	12	10	6-13	4	6	2-10
9	7-11	13	11	7-13	5	7	3-11
10	8-12	Not Available			6	8	4-12
11	9-13	Not Available			7	9	5-13
12	10-13	Not Available			8	10	6-13
13	11-13	Not Available			9	11	7-13

Tabla 7 - Uso del ancho de banda de 40MHz en la banda de los 2,4 GHz

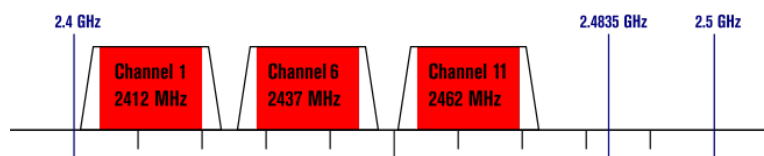
De todas formas, el uso de 802.11n en el rango de los 2,4GHz sigue teniendo el problema de las interferencias y los solapamientos tal y como ocurre en los otros estándares anteriores. Para ello, los canales que quedan definidos sin solapamiento son los siguientes⁹:

Non-Overlapping Channels for 2.4 GHz WLAN

802.11b (DSSS) channel width 22 MHz



802.11g/n (OFDM) 20 MHz ch. width - 16.25 MHz used by sub-carriers



802.11n (OFDM) 40 MHz ch. width - 33.75 MHz used by sub-carriers

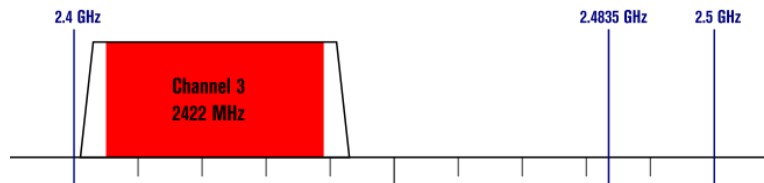


Ilustración 7 - Canales sin solapamiento de 802.11 b/g/n en los 2,4 GHz

⁹ http://en.wikipedia.org/wiki/List_of_WLAN_channels

Este estándar está tomando mucha fuerza y sufriendo un crecimiento muy rápido, ya que los operadores que dan acceso a Internet ya montan sus *routers* con este estándar integrado. Además, como las conexiones que se están empezando a tener en las casas son de fibra óptica, la velocidad que se obtiene en el enlace *wireless* se asemeja mucho a la que nos proporciona el proveedor de servicios en la parte de Internet. Con esto, los dispositivos *wireless* no sufren pérdida de calidad frente a los equipos conectados por cable directamente al *router*.

3.5.- ESTÁNDAR 802.11E

El estándar 802.11e define una serie de cambios en la capa MAC del protocolo 802.11 para poder soportar QoS y permitir así la mejora de calidad para servicios multimedia que requieran transportar voz, audio o vídeo. En el modelo original se definen dos capas llamadas DCF (*Distributed Coordination Function*) y PCF (*Point Coordination Function*).

DCF se encarga de coordinar el diálogo entre las estaciones emisoras de *wireless*. Para ello utiliza un algoritmo en el que si una estación desea emitir, lo primero que ha de hacer es escuchar al canal durante un tiempo que se denomina DIFS (*DCF Interframe Space*). Si durante este tiempo el canal está libre puede emitir. Para calcular este tiempo se usa la siguiente fórmula:

$$DIFS = SIFS + (2 * Slot Time)$$

Ecuación 1 - Cálculo de tiempo DIFS

Donde SIFS (*Short Interframe Space*) es el tiempo mínimo que pasa entre una trama de datos y su paquete de reconocimiento (*acknowledgment*). A continuación se ponen dos tablas donde se relacionan estos parámetros y sus valores habituales¹⁰¹¹:

Standard	Slot time (µs)	DIFS (µs)
IEEE 802.11-1997 (FHSS)	50	128
IEEE 802.11-1997 (DSSS)	20	50
IEEE 802.11b	20	50
IEEE 802.11a	9	34
IEEE 802.11g	9 or 20	28 or 50
IEEE 802.11n (2.4 GHz)	9 or 20	28 or 50
IEEE 802.11n (5 GHz)	9	34
IEEE 802.11ac (5 GHz)	9	34

Tabla 8 - Valores de DIFS según la norma usada

Standard	SIFS (µs) ¹¹
IEEE 802.11-1997 (FHSS)	28
IEEE 802.11-1997 (DSSS)	10
IEEE 802.11b	10
IEEE 802.11a	16
IEEE 802.11g	10
IEEE 802.11n (2.4 GHz)	10
IEEE 802.11n (5 GHz)	16
IEEE 802.11ac	16

Tabla 9 - Valores de SIFS según la norma usada

De todas formas, el DCF tiene ciertas limitaciones. Las más reseñables son que no tiene ningún tipo de QoS implementada y si varias estaciones quieren transmitir a la vez se producen muchas colisiones en la red.

¹⁰ http://en.wikipedia.org/wiki/DCF_Interframe_Space

¹¹ <http://en.wikipedia.org/wiki/SIFS>

Por su parte, PCF es un mecanismo de coordinación en un modo de *wireless* llamado “infraestructura”, donde todas las estaciones se conectan a un AP (*Access Point*) que es el encargado de coordinar la comunicación de las estaciones. El AP siempre espera un tiempo PIFS (*PCF Interframe Space*) para captar el canal. Este tiempo es menor que el DIFS, aún así, un AP siempre tiene prioridad sobre las estaciones. Este tiempo se calcula con la siguiente fórmula:

$$PIFS = SIFS + Slot Time$$

Ecuación 2 - Cálculo del tiempo PIFS

Siempre se cumple que $DIFS > PIFS > SIFS$. En la siguiente tabla se muestran los valores de PIFS según el estándar que estemos usando¹²:

Standard	Slot time (μs)	PIFS (μs)
IEEE 802.11-1997 (FHSS)	50	78
IEEE 802.11-1997 (DSSS)	20	30
IEEE 802.11b	20	30
IEEE 802.11a	9	25
IEEE 802.11g	9 or 20	19 or 30
IEEE 802.11n (2.4 GHz)	9 or 20	19 or 30
IEEE 802.11n (5 GHz)	9	25
IEEE 802.11ac	9	25

Tabla 10 - Valores de PIFS según la norma usada

En PCF, los APs mandan unas tramas especiales llamadas *beacons*, las cuales llevan información acerca del SSID (*Service Set Identifier*) de la red, el número de canal y el tipo de seguridad que se usa. Estas tramas se envían en intervalos regulares (generalmente cada 0,1s). Entre cada trama *beacon*, PCF define dos periodos, CFP (*Contention Free Period*) y el CP (*Contention Period*). Durante CP se usa DCF. Durante el CFP el AP envía paquetes CF-Poll (*Contention-Free-Poll*) a cada estación, uno cada vez, con el que da permiso para enviar un paquete. Aunque esto es una mejora para la integración de QoS, como no define clases de tráfico que es algo común en QoS, no es una solución aceptable.

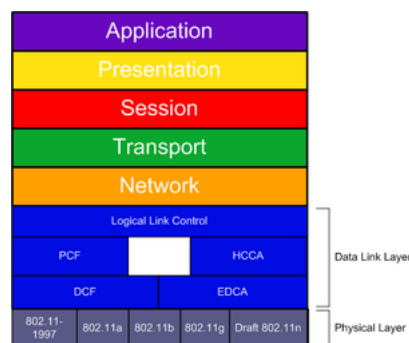


Ilustración 8 - Encaje de 802.11 dentro del modelo OSI

En la imagen anterior¹³ se ve la relación entre el DCF y el PCF con la nueva estructura que propone el 802.11e, que es EDCA (*Enhanced distributed channel Access*) y HCCA (*HCF Controlled Channel Access*). Ambos se definen en el HCF (*Hybrid Coordination Function*), que es la nueva manera que tiene este nuevo estándar para implementar el QoS.

En EDCA se define que un tráfico prioritario será enviado antes que un tráfico menos prioritario, es decir, una estación que tiene tráfico prioritario esperará menos tiempo para mandar sus paquetes que una estación con tráfico menos prioritario. Para poder definir el tiempo que tiene, se usa el AIFS (*Arbitration*

¹² http://en.wikipedia.org/wiki/PCF_Interframe_Space

¹³ http://en.wikipedia.org/wiki/IEEE_802.11e

Inter-Frame Space) para los paquetes prioritarios. Los valores exactos de este tiempo dependen de la capa física que se use para transmitir los datos. Para calcular este tiempo hay que utilizar la fórmula:

$$AIFS = AIFSN[AC] * Slot Time + SIFS$$

Ecuación 3 - Cálculo del tiempo AIFS

Donde AIFSN es el número de AIFS que depende de la AC (*Access Categories*), que son los niveles de prioridad de tráfico que se definen en EDCA.

EDCA provee un periodo de acceso al canal llamado TXOP (*Transmit Opportunity*). Este es un intervalo de tiempo limitado en el cual una estación puede enviar tantas tramas como pueda. Un tiempo de TXOP de cero implica que está limitado a un único MSDU (*MAC Service Data Unit*) o un MMPDU (*MAC Management Protocol Data Unit*).

Otro parámetro que se define en EDCA es el de CW (*Contention Window*), que se define según el tipo de tráfico que se espera tener en cada AC. Los valores de CWmin y CWmax se definen según los valores de aCWmin y aCWmax que dependen de la capa física que se utilice. En la siguiente tabla¹⁴ se muestra cómo se relacionan estos conceptos y los nombres que se asignan a cada AC:

AC	CWmin	CWmax
Background (AC_BK)	aCWmin	aCWmax
Best Effort (AC_BE)	aCWmin	aCWmax
Video (AC_VI)	(aCWmin+1)/2-1	aCWmin
Voice (AC_VO)	(aCWmin+1)/4-1	(aCWmin+1)/2-1

Tabla 11 - Cálculo de los límites del parámetro CW

Un ejemplo para valores típicos de aCWmin=15 y aCWmax=1023 daría los siguientes resultados que, además, son los valores por defecto de EDCA:

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

Tabla 12 - Valores por defecto de EDCA para cada AC

La otra capa que define 802.11e es la de HCCA, la cual funciona de una manera muy parecida a PCF. La mayor diferencia es que en PCF sólo se tienen los tiempos de CFP y CP entre dos tramas de *beacon*, mientras que en HCCA se permite usar CFPs en casi cualquier CP. A este tipo de CFP se le llama CAP (*Controlled Access Phase*) y es iniciada por un AP cuando quiere enviar una trama a una estación o recibirla desde ella de una manera libre de contención. De hecho, la CFP es también una CAP. Durante una CAP, el HC (*Hybrid Coordinator*) -el cual también es el AP- controla el acceso al medio.

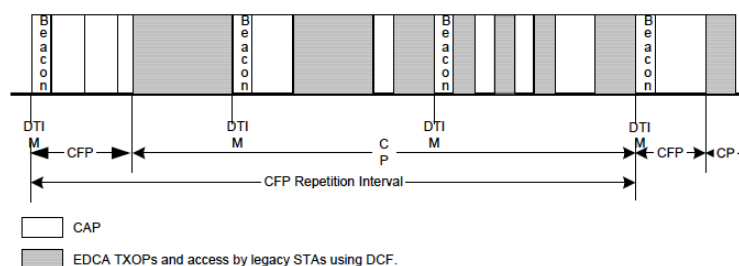


Ilustración 9 - Ejemplo de generación de los CAPs durante los CP

¹⁴ http://en.wikipedia.org/wiki/IEEE_802.11e

La otra diferencia con PCF es que se definen el TC (*Traffic Class*) y el TS (*Traffic Streams*). Esto significa que el HC no está limitado a una cola por estación y puede ofrecer un tipo servicio por sesión. El HC también puede elegir cualquier manera para coordinar estos flujos o sesiones (no sólo en modo *round-robin*, que es de uno en uno). Es más, las estaciones dan información de la longitud de sus colas para cada TC. El HC puede usar esta información para dar prioridad a una estación sobre otra, o para ajustar mucho mejor sus mecanismos de programación. Otra diferencia es que a las estaciones se les asigna un TXOP, así pueden enviar múltiples paquetes de una vez durante un tiempo seleccionado por el HC. Durante un CFP, el HC permite a la estaciones enviar datos mandándoles tramas de CF-Poll.

A la HCCA se le considera generalmente la función de coordinación más avanzada (y compleja). Con HCCA el QoS se puede configurar de una manera muy precisa. Las estaciones con QoS habilitado tienen la capacidad de solicitar parámetros específicos para la transmisión (tasa de transferencia, *jitter*, etc.) los cuales permiten un funcionamiento mucho más eficiente de aplicaciones como VoIP o vídeo sobre redes WiFi.

4.- PRESENTACIÓN DEL MODELO TEÓRICO PARA LA SIMULACIÓN

4.1.- INTRODUCCIÓN

Los autores del artículo presentan un modelo sobre el cual desarrollar y simular el comportamiento de una red WiFi que soporta el protocolo 802.11e y que transporta tráfico VoIP. Este tipo de tráfico ya se ha comentado que es muy sensible a los retardos. Este hecho lleva a la pregunta de cuál es la capacidad de llamadas que se tienen en este tipo de redes usando diferentes velocidades que dan los protocolos de la familia 802.11. Asimismo, se hace uso de tres tipos de codecs de audio que son el G.711, G.729 y G.723.1. Cada uno de ellos tiene una codificación y tasa de bits diferentes tal y como se ha visto en los apartados anteriores.

Para este modelo, y en general, una llamada de VoIP consiste en una comunicación bidireccional desde una de las estaciones hasta el QAP (*QoS Access Point*) o a la inversa. Estos QAP son los encargados de realizar las funciones del HC que se han mencionado anteriormente al describir el protocolo 802.11e.

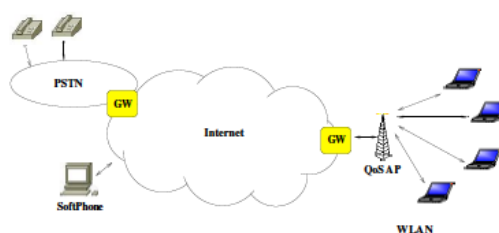


Ilustración 10 - Ejemplo de red WiFi para VoIP

4.2.- PARÁMETROS DE LA SIMULACIÓN

4.2.1- E-MODEL

Para poder hacer las mediciones, no sólo se basan en el número de llamadas que se puedan lanzar por el ancho de banda proporcionado por la red, si no que además, estas llamadas tienen que tener calidad suficiente. Para ello hacen uso del *E-Model* definido en la recomendación G.107 del ITU-T para la medición de calidad de audio.

El *E-Model* ha demostrado ser útil como herramienta de planificación de la transmisión para la evaluación de los efectos combinados de las variaciones en varios parámetros de transmisión que afectan a la calidad de la conversación de telefonía auricular en 3,1 KHz. Este modelo computacional se puede utilizar, por ejemplo, por los planificadores de transporte para ayudar a asegurar que los usuarios estarán satisfechos con el rendimiento de transmisión de extremo a extremo, mientras que evita el exceso de ingeniería de redes. Se debe enfatizar que la salida principal del modelo es el "factor de calificación" R pero esto puede ser transformado para dar estimaciones de la opinión de los clientes. Estas estimaciones se hacen solamente con fines de planificación de transmisión y no para predecir la opinión del cliente real.

El *E-Model* se puede utilizar con confianza para muchas combinaciones de gran importancia para los planificadores de transporte, pero para algunas combinaciones de parámetros de gran importancia, las predicciones del *E-Model* han sido cuestionadas y actualmente se encuentran en estudio.

El *E-Model* hace uso del factor R donde los valores que toma van desde 0 hasta 100, siendo 0 el peor de los casos y 100 el mejor de ellos. Para calcular el factor R se usa la siguiente fórmula:

$$R = R_0 - I_s - I_d - I_e + A$$

Ecuación 4 - Cálculo del factor R

Donde R_0 agrupa el efecto del ruido, I_s incluye los efectos de alteraciones relacionadas con la cuantificación de la señal de voz (en este caso su valor será de cero), I_d representa las alteraciones causadas por los retardos, I_e cubre las alteraciones causadas por las bajas tasas de transmisión de los codecs y los paquetes perdidos. El factor positivo A compensa las alteraciones anteriores bajo varias condiciones definidas por el usuario. En el caso de la telefonía móvil A tiene un valor de 10. En el caso de la VoIP A va a tener un valor de 0.

4.2.2- MOS (MEAN OPINION SCORE)

Además, se usa la medición subjetiva MOS (*Mean Opinion Score*), ya que las mediciones objetivas no tienen en cuenta el retardo en la voz para transmisiones en tiempo real. MOS es una prueba que se ha utilizado durante décadas en las redes de telefonía para obtener la opinión de los usuarios humanos para determinar la calidad de la red. Históricamente MOS era una medida subjetiva, donde los oyentes se sentaban en una habitación "tranquila" y puntuaban la calidad de la llamada tal y como la percibían; para la recomendación P.800 del ITU-T, "El hablador debe estar sentado en una habitación tranquila con un volumen de entre 30 y 120 m³ y un tiempo de reverberación de menos de 500 ms (preferiblemente en el intervalo de 200-300 ms). El nivel de ruido en las habitaciones debe estar por debajo de 30 dBA sin picos dominantes en el espectro ". La medición de VoIP es más objetiva, y es un cálculo basado en el rendimiento de la red IP sobre la que se transporta la voz. Este cálculo se define en el estándar P.862 PESQ del ITU-T. Como la mayoría de los estándares, la implementación está algo abierta a la interpretación del fabricante de equipos o de *software*. Por otra parte, debido a los avances tecnológicos de los fabricantes de teléfonos, un MOS calculado de 3.9 en una red VoIP en realidad puede sonar mejor que la puntuación anteriormente subjetiva de >4.0.

En multimedia (audio, telefonía de voz, o vídeo), especialmente cuando se utilizan los codecs para comprimir la voz y disminuir el requisito de ancho de banda (por ejemplo, de una conexión de voz digitalizada a partir de la norma de modulación de 64 kilobits/segundo PCM), el MOS proporciona una indicación numérica de la calidad percibida de la comunicación recibida después de la compresión y/o la transmisión desde la perspectiva de los usuarios. La MOS se expresa como un solo número en el rango de 1 a 5, donde 1 es la calidad más baja de audio percibida, y 5 es la medición de la calidad de audio más alta percibida.

Las pruebas de voz para MOS están especificadas por el ITU-T en la recomendación P.800.

Codec	Data rate [kbit/s]	Mean opinion score (MOS)
G.711 (ISDN)	64	4.1
iLBC	15.2	4.14
AMR	12.2	4.14
G.729	8	3.92
G.723.1 r63	6.3	3.9
GSM EFR	12.2	3.8
G.726 ADPCM	32	3.85
G.729a	8	3.7
G.723.1 r53	5.3	3.65
G.728	16	3.61
GSM FR	12.2	3.5

Tabla 13 - Ejemplo de puntuaciones MOS de algunos codecs

4.2.3- RELACIÓN ENTRE E-MODEL Y MOS

El factor R puede ser traducido a la escala MOS a través de la siguiente expresión:

$$MOS = 1 + 0,035R + 7 * 10^{-6}R(R - 60)(100 - R)$$

Ecuación 5 - Traducción de Factor R a escala MOS

En la siguiente tabla se muestran las diferentes clases de calidad y sus correspondientes valores de MOS y factor R:

Calidad de Transmisión de la Voz	MOS	Factor R
Best	4,50 - 4,34	100 - 90
High	4,34 - 4,03	90 - 80
Medium	4,03 - 3,60	80 - 70
Low	3,60 - 3,10	70 - 60
Poor	3,10 - 2,58	60 - 50

Tabla 14 - Valores de MOS y Factor R para la calidad de transmisión de voz

La calidad de la voz cae en el rango de 70, por lo que R=70 (MOS=3,6) será el valor de referencia para evaluar la capacidad en este estudio. Este será el valor mínimo que debería de cumplir un códec para determinar que una llamada se realiza con cierta calidad.

4.2.4- PARÁMETROS PARA EL ESTÁNDAR 802.11

Para el caso de la red *wireless*, una llamada de VoIP es una comunicación bidireccional, y la capacidad de las llamadas será el número máximo de conexiones bidireccionales que un AP pueda soportar. Para ello se considera que para el modelo de VoIP se use CBR (*Constant Bit Rate*) para los codecs elegidos. Los parámetros usados para el modelo son los siguientes:

B	Beacon frame size (320 bits)
H	Protocol header overhead (456 bits)
P	Physical layer header (192 bits)
T _B	Beacon interval (0,01 sec)
SIFS	SIFS time interval (28 μs)
CF _{end}	CFP end frame (192 bits)

Tabla 15 - Parámetros usados para el modelado del sistema

Estos serán los parámetros que se usarán para los cálculos en las fórmulas para el cálculo del número de llamadas que se explican a continuación.

4.3.- CÁLCULO DEL NÚMERO DE LLAMADAS

Para calcular el número de llamadas, hay que considerar el tiempo que se tiene de CFP (T_{CFP}) y el tiempo de CP (T_{CP}). El número total de llamadas simultáneas de VoIP será la suma de las llamadas que se puedan soportar en cada uno de estos periodos. La ecuación para calcular el tiempo más largo posible para una llamada durante el periodo de CFP es:

$$T_{poll-cfp} = \frac{((C * (T_{voice} + T_{CFP}) + H + P) * 2)}{R} + 2 * SIFS$$

Ecuación 6 - Ecuación para el cálculo de $T_{poll-cfp}$

Donde C es la tasa de transferencia a la que el códec está generando los paquetes, T_{voice} es el retardo producido por el códec (generalmente de paquetización), T_{CFP} es el máximo tiempo de *interpoll* para una llamada en CFP. R es la tasa de transmisión de la red *wireless* utilizada y SIFS es el tiempo necesario de separación entre todos los fragmentos y los ACKs.

De la misma manera se puede obtener la ecuación para los periodos de CAP dentro de los CP. En este caso, en vez de usar el tiempo máximo de *interpoll* en CFP, se cambia por el del periodo de CP (T_{CP}):

$$T_{poll-cp} = \frac{((C * (T_{voice} + T_{CP}) + H + P) * 2)}{R} + 2 * SIFS$$

Ecuación 7 - Ecuación para el cálculo de $T_{poll-cp}$

Para poder usar los datos de tasas de los códec y los retardos, se toman los valores de la siguiente tabla:

CODEC	Bit Rate (Kbit/s)	Retardo (ms)
G.711	64	20
G.729	8	20
G.723.1	6,3	30

Tabla 16 - Parámetros de los codecs para retrasos y tasas de transferencia

Además hay que calcular el tiempo de overhead relacionado con el periodo de CFP, el cual es el tiempo necesario para el envío de las señales de Beacon y CF_{END} . Este tiempo se calcula de la siguiente manera:

$$T_{overhead} = \left(\frac{B + P}{R} + SIFS \right) * \left(\frac{T_{SF} - T_{CP}}{T_B} \right) + \frac{CF_{end} + P}{R}$$

Ecuación 8 - Ecuación para el cálculo de $T_{overhead}$

Ahora ya se puede calcular el número de llamadas en el periodo de CFP:

$$N_{poll-cfp} = \frac{T_{CFP} - T_{overhead}}{T_{poll-cfp}}$$

Ecuación 9 - Ecuación para el cálculo del total de llamadas en un periodo de CFP

Y en el periodo de CP:

$$N_{poll-cp} = \frac{T_{CP}}{T_{poll-cp}}$$

Ecuación 10 - Ecuación para el cálculo del total de llamadas en un periodo de CP

Y el número total de llamadas será la suma de ambos valores.

4.4.- HERRAMIENTA USADA PARA LA SIMULACIÓN

Para reproducir el modelo utilizado en el artículo se elige la herramienta Matlab. Con ella se pretende conseguir las mismas gráficas que el autor y así demostrar la validez de las ecuaciones propuestas. Se ha

elegido esta herramienta por ser la más utilizada en entornos de investigación. La potencia y sencillez de esta herramienta permiten realizar muchas más funciones de las que se van a utilizar para esta simulación.

Además, sus asistentes para pintar las gráficas permiten poder hacer cambios de escalas en las leyendas de los ejes de coordenadas sin que ello afecte a los cálculos. Gracias a todas estas herramientas se ha podido hacer un desarrollo bastante rápido de la simulación y con un buen resultado.

MATLAB (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje *M*) y servicio de especie. Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

MATLAB es un entorno de cálculo técnico de altas prestaciones para cálculo numérico y visualización. Integra:

- Análisis numérico
- Cálculo matricial
- Procesamiento de señales
- Gráficos

Es un entorno fácil de usar, donde los problemas y las soluciones son expresados como se escriben matemáticamente, sin la programación tradicional. MATLAB fue escrito originalmente para proporcionar un acceso sencillo al software matricial desarrollado por los proyectos LINPACK y EISPACK, que juntos representan lo más avanzado en programas de cálculo matricial. MATLAB es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento. Esto permite resolver muchos problemas numéricos en una fracción del tiempo que llevaría hacerlo en lenguajes como C, BASIC o FORTRAN. MATLAB ha evolucionado en los últimos años a partir de la colaboración de muchos usuarios. En entornos universitarios se ha convertido en la herramienta de enseñanza estándar para cursos de introducción en álgebra lineal aplicada, así como cursos avanzados en otras áreas. En la industria, MATLAB se utiliza para investigación y para resolver problemas prácticos de ingeniería y matemáticas, con un gran énfasis en aplicaciones de control y procesamiento de señales. MATLAB también proporciona una serie de soluciones específicas denominadas *TOOLBOXES*. Estas son muy importantes para la mayoría de los usuarios de MATLAB y son conjuntos de funciones MATLAB que extienden el entorno MATLAB para resolver clases particulares de problemas como:

- Procesamiento de señales
- Diseño de sistemas de control
- Simulación de sistemas dinámicos
- Identificación de sistemas
- Redes neuronales y otros.

Probablemente la característica más importante de MATLAB es su capacidad de crecimiento. Esto permite convertir al usuario en un autor contribuyente, creando sus propias aplicaciones. En resumen, las prestaciones más importantes de MATLAB son:

- Escritura del programa en lenguaje matemático.
- Implementación de las matrices como elemento básico del lenguaje, lo que permite una gran reducción del código, al no necesitar implementar el cálculo matricial.
- Implementación de aritmética compleja.
- Un gran contenido de órdenes específicas, agrupadas en *TOOLBOXES*.
- Posibilidad de ampliar y adaptar el lenguaje, mediante ficheros de script y funciones .m.

MATLAB provee acceso inmediato a las características gráficas especializadas requeridas en ingeniería y ciencias. Potente herramienta para pintar gráficas orientada a objetos gráficos le permite pintar los resultados de su análisis, incorporar gráficos en sus modelos de sistemas, rápidamente presentar complejos 3-D objetos, y crear resultados de presentación, entre lo cual se destaca:

- Representaciones 2-D y 3-D, incluyendo datos triangulados y reticulados
- Representaciones 3-D quiver, ribbon, y stem
- Control de fuentes, letras Griegas, símbolos, subíndices y superíndices
- Selección expandida de símbolos marcadores de curvas
- Gráficos de torta, de barras 3-D y gráficos de barras horizontales
- Gráficos 3-D y sólido modelado
- Representación de imágenes y archivos I/O
- Gráficos comentados
- Leer/Escribir archivos de datos Hierarchical Data Format (HDF)
- Presentación de OpenGL software y hardware
- Animación
- Display de buffer x rápido y exacto
- Soporte de colores verdaderos (24-bit RGB)
- Fuentes múltiples de luz para superficies coloreadas
- Vista basada en cámara y control de perspectiva
- Iluminación Plana, Gouraud y Phong
- Soporte eficiente de imagen de datos de 8-bit
- Control de eje y cámara
- Propiedades de superficie y patch
- Modelos de iluminación
- Control gráfico de objetos
- Impresión y representación de copias
- Formatos gráficos exportables
- Soporte de publicación de escritorio

5.- RESULTADOS OBTENIDOS SEGÚN EL MODELO

5.1.- INTRODUCCIÓN

Los autores del artículo pretenden conseguir con este estudio crear un modelo analítico que permita estudiar el número de llamadas que se pueden realizar en una red *wireless*. Para ello hacen uso de los tres codecs que se han presentado anteriormente junto con las nuevas características que aporta el estándar 802.11e.

Este nuevo estándar introduce el concepto de los CAPs (*Controlled Access Periods*) dentro de los CP (*Contention Period*). Estos nuevos tiempos permiten sondear los dispositivos desde los AP para así ver si tienen información que enviar. Así se consigue un mejor aprovechamiento del tiempo de envío de información, y se consigue aumentar el número de llamadas que se pueden realizar.

Con las medidas que se presentan a continuación se verá cómo lo que pretenden los autores se puede conseguir gracias a estas nuevas características y los resultados tan espectaculares que se consiguen.

5.2.-CÁLCULO DE MOS EN FUNCIÓN DEL RETARDO

Tal y como hacen los autores, la primera gráfica que se obtiene es la relación entre el efecto del retardo en los codecs utilizados y su valor de MOS. Para ello se usan las ecuaciones mencionadas en el apartado anterior, obteniendo la siguiente gráfica:

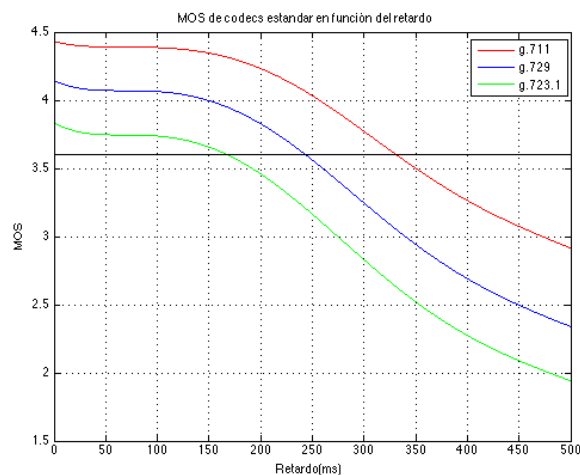


Ilustración 11 - MOS de codecs típicos de VoIP en función del retardo de la red

Tal y como comentan los autores en su artículo, los valores del retardo para los cuales cada códec alcanza el valor de MOS=3,6 son de 170 ms para G.723.1, 240 ms para G.729 y 325 ms para G.711. Esto viene a demostrar lo que se venía diciendo de que el códec G.711 es menos sensible a los retardos de la red que los otros dos codecs.

5.3.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO DCF/PCF

Tras obtener esta gráfica, se realiza la que corresponde al número máximo de llamadas que puede tener una red *wireless* usando el estándar 802.11b (11 Mbps) y usando el modo de trabajo DCF/PCF. Para ello se ha usado el modelo del trabajo realizado por Malathi Veeraraghavan, Nabeel Cocker y Tim Moors en

[32], pero adaptado al modelo del autor del artículo en el que se basa esta simulación. La modificación que se realiza para poder realizar las gráficas en modo DCF/PCF es la siguiente:

$$T_{poll-cfp} = \frac{\left(\left((C * (T_{voice} + T_{SF})) + H + P \right) * 2 \right)}{R} + 2 * SIFS$$

Ecuación 11 - Cálculo del tiempo de CFP para el modo DCF/PCF

Aquí se ve claramente que al retardo de paquetización del códec se le suma el tiempo de la *superframe* en vez del tiempo de CFP. Con este cambio en la ecuación se obtiene la siguiente gráfica, donde se ve el número máximo de llamadas que se pueden procesar dependiendo del tamaño que usemos de *superframe*:

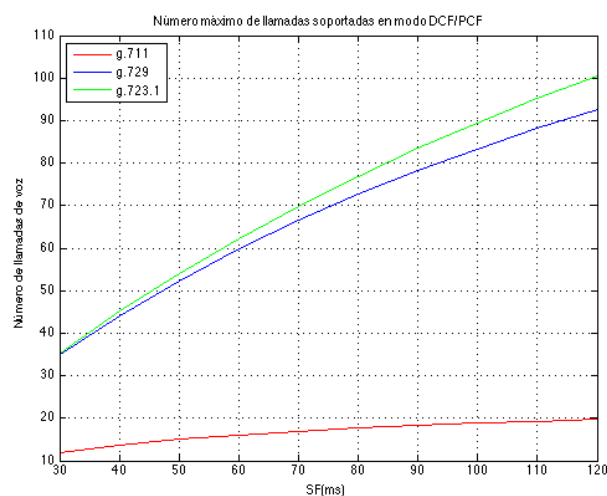


Ilustración 12 - Número máximo de llamadas soportadas en DCF/PCF

También se demuestra con esta gráfica que, dependiendo del tiempo que se le asigne a la *superframe*, afecta de manera directa al número de llamadas que se pueden tener en la red *wireless*. Esto es obvio ya que al aumentar el tiempo de la *superframe* también estamos aumentando el tiempo del CFP, que es el periodo en el que en este modo de trabajo podemos realizar las llamadas.

Asimismo se observa la gran diferencia entre las llamadas que se pueden cursar con el códec G.711 y los otros dos. En el tiempo máximo de *superframe* para cada códec tenemos que G.711 puede cursar 20 llamadas, G.729 puede 95 y G.723.1 puede 107.

5.4.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA

Una vez obtenida esta gráfica se realiza la misma operación pero usando el modo de trabajo EDCA/HCCA tal y como se plantea en el artículo. En este caso se verá como el número de llamadas que se puede realizar con cada códec es bastante mayor ya que, dentro del periodo de CP, donde antes no se podían realizar llamadas, ahora sí las tenemos gracias a los CAPs.

Al igual que en la gráfica anterior, se varía el tiempo de *superframe* para ver cómo afecta la número máximo de llamadas que se pueden procesar. Con esto queda la siguiente gráfica:

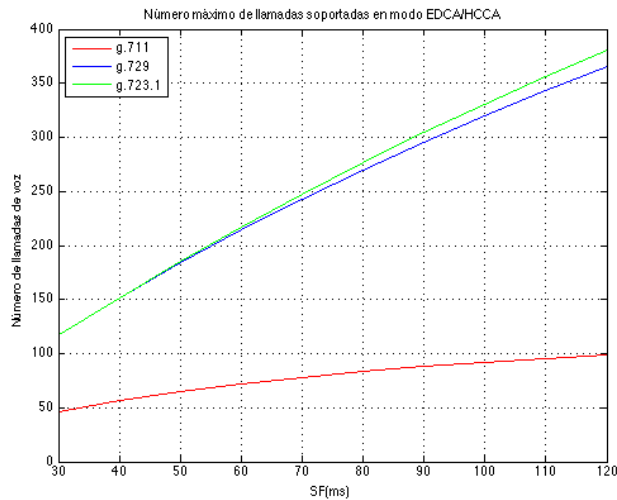


Ilustración 13 - Número máximo de llamadas en el modo EDCA/HCCA

Al igual que en la gráfica en el modo DCF/PCF, se observa cómo se incrementa el número de llamadas a medida que se aumenta el tiempo de la *superframe*. Pero en este caso se ve que al trabajar en el nuevo modo, el número de llamadas se ha incrementado mucho. Este se debe a que ahora también se pueden cursar llamadas en los CP, tal y como se ha explicado antes.

En esta gráfica, para los valores máximos de *superframe*, las llamadas que se pueden realizar con cada códec son: 99 para G.711, 360 para G.729 y 392 para G.723.1. Como se puede ver, el incremento es bastante notable.

5.5.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA VARIANDO EL TIEMPO DE CP

Una vez obtenidos estos datos, la siguiente gráfica que nos proponen los autores es ver cuántas llamadas se pueden realizar dejando el tiempo de SF (*SuperFrame*) fijo a 120 ms mientras que lo que se varía es el tiempo de CP (*Contention Period*). Con esta premisa, la gráfica que se obtiene es la siguiente:

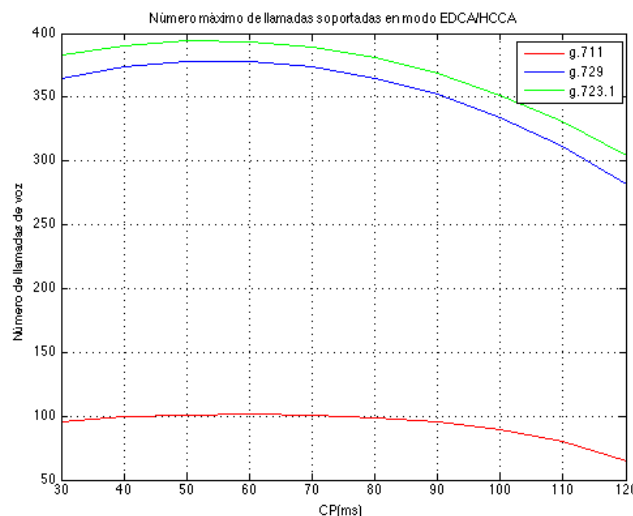


Ilustración 14 - Número máximo de llamadas que se pueden realizar mientras se incrementa el tiempo de CP

Tal y como comentan en su artículo, el máximo número de llamadas se produce en el intervalo entre los 50 ms y los 60 ms. En la gráfica obtenida para reproducir su modelo se ha obtenido el mismo resultado. Con este dato se recalca que simplemente ajustando adecuadamente los valores de CP y CFP, se puede

conseguir un incremento considerable en el número de llamadas que se pueden realizar. Esto es así porque durante el periodo de CP se desperdicia algún tiempo, con lo que si lo ajustamos al máximo que se observa en esta gráfica, podemos obtener un mejor rendimiento de la red.

También se demuestra que aunque los CAPs dentro del periodo de CP permite aumentar mucho el número de llamadas simultáneas, esto no implica que si se aumenta mucho este tiempo siempre se obtendrá un buen resultado. Así queda que sólo en el intervalo entre 50 y 60 ms es donde se consigue el máximo de llamadas posibles.

5.6.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS SEGÚN LA VELOCIDAD DE TRANSMISIÓN

Para finalizar el artículo, los autores realizan una comparativa del número de llamadas máximas que se pueden realizar según la tasa de transferencia que se use, yendo desde los 2 Mbps hasta los 54 Mbps y manteniendo el tiempo de SF fijo en 120 ms. Además, compara entre el modo de funcionamiento DCF/PCF frente al modo EDCA/HCCA. Con ello se consiguen las siguientes gráficas:

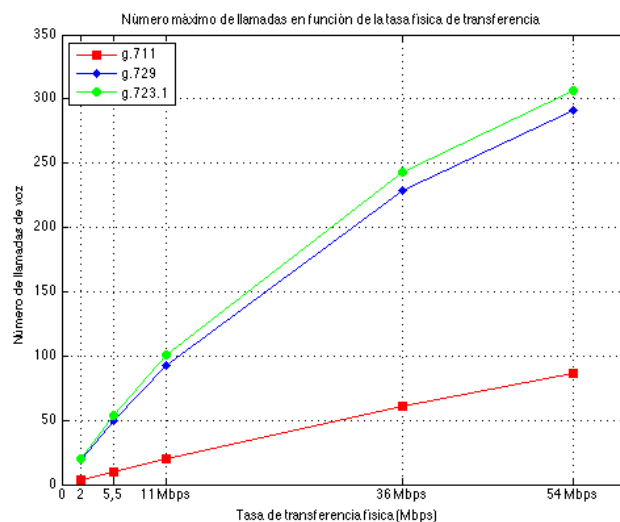


Ilustración 15 - Número máximo de llamadas soportado en modo DCF/PCF en función de la tasa de transferencia

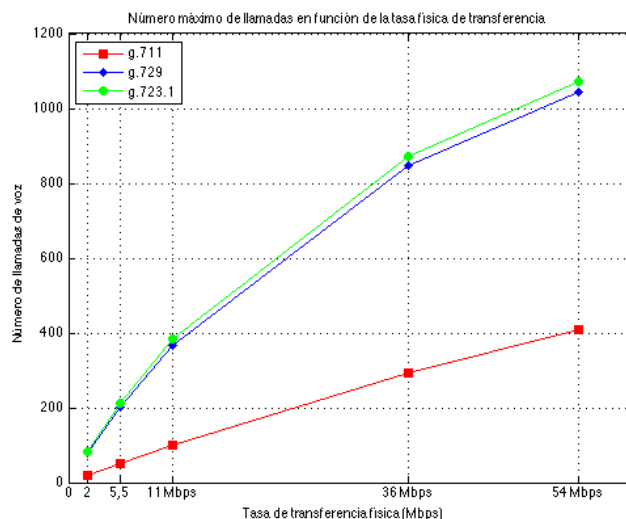


Ilustración 16 - Número máximo de llamadas soportado en modo EDCA/HCCA en función de la tasa de transferencia

Con estas gráficas los autores demuestran cómo se incrementa el número de llamadas en función del ancho de banda del que se disponga. Además, se ve claramente cómo es mucho mayor el número de llamadas en el modo EDCA/HCCA frente al modo DCF/PCF. Además de las gráficas, se presentan unas tablas resumen sobre estas dos gráficas, las cuales se muestran a continuación:

Physical Rate	G.711 (64 Kbps)	G.729 (8 Kbps)	G.723.1 (6,3 Kbps)
2 Mbps	4	19	46
5,5 Mbps	10	51	58
11 Mbps	20	95	107
36 Mbps	61	232	254
54 Mbps	87	295	318

Tabla 17 - Número máximo de llamadas para los codecs estándar en función de la tasa de transmisión y usando el modo de funcionamiento DCF/PCF

Physical Rate	G.711 (64 Kbps)	G.729 (8 Kbps)	G.723.1 (6,3 Kbps)
2 Mbps	19	77	86
5,5 Mbps	51	199	219
11 Mbps	99	360	392
36 Mbps	288	813	861
54 Mbps	400	997	1045

Tabla 18 - Número máximo de llamadas para los codecs estándar en función de la tasa de transmisión y usando el modo de funcionamiento EDCA/HCCA

A la luz de todos estos resultados, se ve que simplemente variando los tiempos de SF y CP y aplicando el modo de funcionamiento EDCA/HCCA, se pueden conseguir un gran número de llamadas simultáneas usando el mismo ancho de banda.

5.7.- ACLARACIONES DEL APARTADO

Para poder realizar la simulación del modelo propuesto en el artículo usado, se han tenido que realizar muchas pruebas, ya que el autor no deja escrito qué valores de tiempo usa para los CP y CFP. Esto ha resultado en un gran problema a la hora de poder obtener los mismos resultados que el autor. Debido a este inconveniente, se han tenido que ajustar algunas de las fórmulas para poder obtener los mismos resultados. No se ha pretendido en ningún momento el copiar exactamente las gráficas del artículo, si no poder obtenerlas con un código realizado en Matlab.

Además del ajuste que se ha tenido que realizar con los tiempo de CFP y CP comentados anteriormente, también se han añadido los retardos de preanálisis de los codecs G.729 y G.723.1 en los parámetros del retardo de paquetización. Sin ellos habría sido imposible conseguir los mismos resultados que el autor.

Estas aclaraciones se pueden verificar viendo el código usado y puesto en el anexo I del presente TFC.

6.- RESULTADOS OBTENIDOS CON LA AMPLIACIÓN DEL MODELO

6.1.- INTRODUCCIÓN

Para ampliar el estudio del modelo original se han realizado algunas modificaciones. En primer lugar se han basado los cálculos sobre el estándar 802.11g (54 Mbps), ya que actualmente es la velocidad más extendida en las redes *wireless*, aunque ya empieza a haber muchos dispositivos que soportan el estándar 802.11n.

En segundo lugar se han incluido, a modo comparativo, las cuatro velocidades máximas que nos da este estándar en la banda de los 2,4 GHz y con un ancho de banda de 20 MHz por canal usando una, dos, tres y cuatro antenas (gracias a la tecnología MIMO)¹⁵. No se ha extendido el estudio a las posibilidades de usar canales de 40 MHz ni a la banda de los 5 GHz por mantener todas las comparativas en la misma línea.

Además se han mantenido las correcciones mencionadas en el apartado anterior para que en las gráficas de comparativa de llamadas según la tasa de transferencia sigan apareciendo los mismos resultados que en la simulación del estudio.

Con todos estos cambios, se han conseguido generar las mismas gráficas que en el apartado anterior para así poder ver las diferencias que nos aportan, sobre todo en el número de llamadas que se pueden procesar al tener un aumento del ancho de banda disponible.

Lo que se ha pretendido con estas modificaciones es ampliar, o mejor dicho, actualizar el modelo presentado en el artículo a las redes *wireless* más usadas en la actualidad. Bien es cierto que el estándar 802.11n está creciendo muy deprisa, pero el estándar g está más implantado por llevar más tiempo en el mercado.

Aún así, no ha sido objetivo de esta ampliación el modificar el modelo propuesto, simplemente el intentar acercar los resultados a las tecnologías usadas actualmente. Por eso los datos que se obtendrán serán comparados con los originales del estudio y ver cómo la evolución de la tecnología aporta mejoras también en este estudio.

6.2.-CÁLCULO DE MOS EN FUNCIÓN DEL RETARDO

En primer lugar se obtiene la gráfica comparando el MOS que se obtiene en función del retardo. En este caso el resultado es exactamente el mismo que en la simulación anterior ya que ninguno de los parámetros modificados afecta a este resultado:

¹⁵ Ver contenido de la tabla 5 de este documento

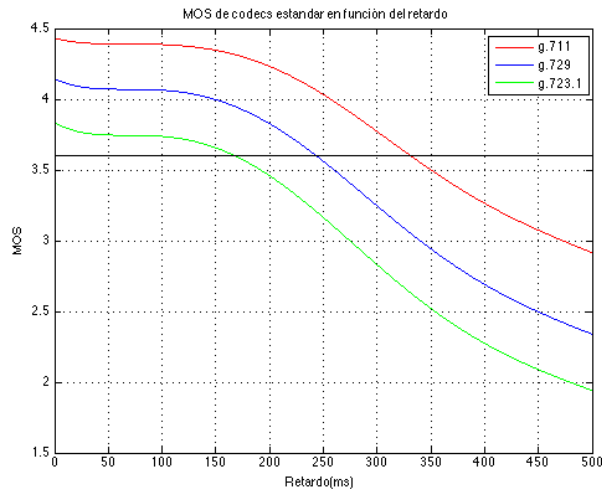


Ilustración 17 - MOS de los codecs estándar en función del retardo

Como ocurría en la simulación anterior, los codecs con un *bit rate* más pequeño, son más propensos a sufrir un deterioro en la calidad a valores de retardo más pequeños frente a codecs con *bit rates* mayores.

Exactamente igual que en la simulación anterior, los valores de los retardos para los cuales los codecs llegan al valor de MOS=3,6 son de 170 ms para G.723.1, 240 ms para G.729 y 325 ms para G.711. Más adelante se verá que, con las nuevas velocidades, es posible que ser menos sensible a los retardos pueda convertirse en una ventaja considerable para G.711.

6.3.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO DCF/PCF

En segundo lugar se obtiene la relación de llamadas máximas que se pueden tener en función del tamaño del tiempo de SF. Como en el caso anterior, el valor de SF varía desde los 30 ms hasta los 120 ms:

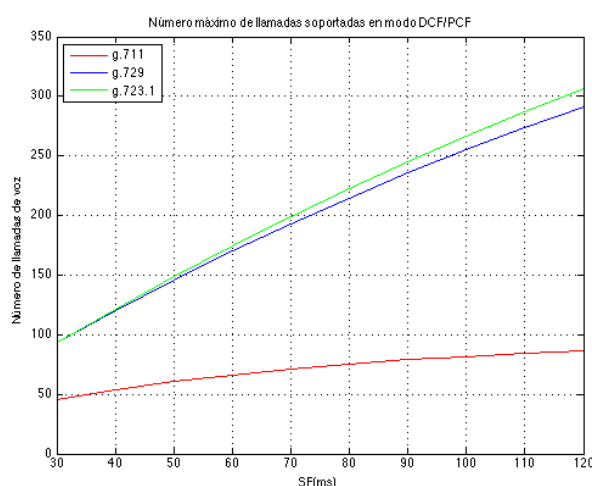


Ilustración 18 - Número de llamadas máximas en función del tiempo de SF en modo DCF/PCF

Tal y como se ve en la gráfica, para este caso el número de llamadas máximas para el códec G.711 con una tamaño de SF de 120 ms es de 87, mientras que para los codecs G.729 y G.723.1 es mucho mayor (igual que en el estudio anterior). En este caso los valores se van a 291 para el códec G.729 y a 307 para el G.723.1. Como se ve son valores bastante mayores que en el estudio anterior ya que disponemos de mucho más ancho de banda.

6.4.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA

Ahora se realiza la misma gráfica pero haciendo uso del modo EDCA/HCCA, para ver, como en el estudio anterior, que el número de llamadas es mucho mayor que en el modo DCF/PCF:

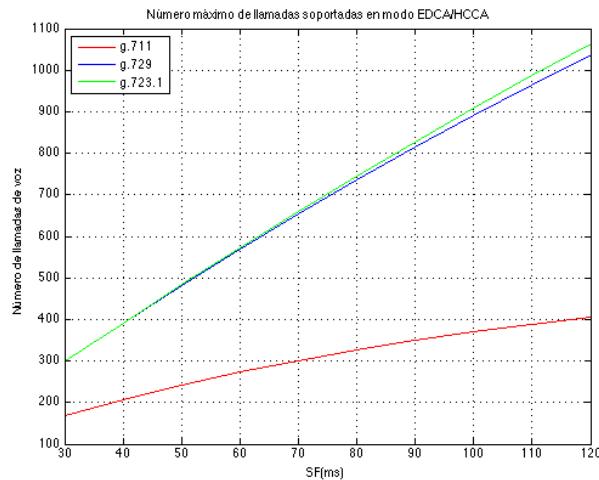


Ilustración 19 - Número de llamadas máximas en función del tiempo de SF en modo EDCA/HCCA

Como en el caso anterior vuelve a ocurrir que el número de llamadas se incrementa muchísimo debido a los CAPs que en el modo DCF/PCF no se tienen. Esto demuestra que se aprovecha mucho mejor el ancho de banda disponible y los tiempos de que se dispone en el nivel de enlace del modelo OSI. Para este caso, el número de llamadas que se tienen para el códec G.711 es de 405 y para los codecs G.729 y G.723.1 son de 1038 y 1064 respectivamente.

Se puede observar también, tanto para el modo DCF/PCF como para EDCA/HCCA, que las llamadas máximas para los codecs G.729 y G.723.1 son muy parecidas. Esto hace pensar que en casi igualdad de condiciones interesa más usar el códec G.729 por tener un MOS más alto y ser menos sensible al retardo.

6.5.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS EN MODO EDCA/HCCA VARIANDO EL TIEMPO DE CP

Si ahora se hace variar el tiempo de CP en vez del de SF, dejando éste fijo a 120 ms y el otro se varía entre 30 ms y 120 ms (al igual que se hacía en el modelo anterior), se obtiene la siguiente gráfica:

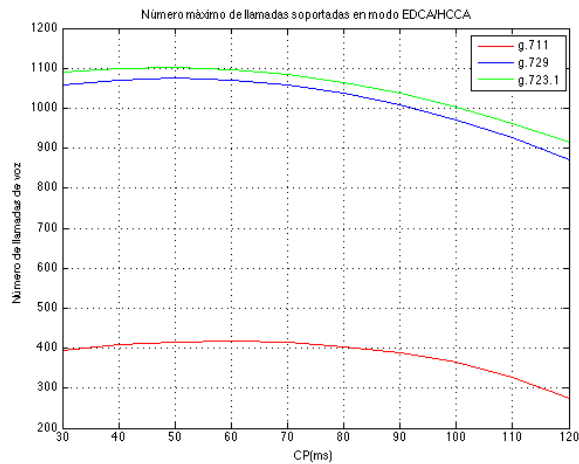


Ilustración 20 - Número máximo de llamadas mientras se incrementa el tiempo de CP en el modo EDCA/HCCA

Como en el caso anterior, el número máximo de llamadas se obtiene entre los 50 ms y los 60 ms. Con esto se vuelve a demostrar que simplemente variando los tiempos de CP y CFP aunque el de SF se mantenga fijo, también podemos obtener un incremento considerable en las llamadas que se pueden tener.

De nuevo se puede observar que hay un tiempo para el periodo de CP que hace que el rendimiento de las llamadas sea mayor. El razonamiento es el mismo que el explicado en la simulación original.

6.6.-CÁLCULO DEL NÚMERO MÁXIMO DE LLAMADAS SEGÚN LA VELOCIDAD DE TRANSMISIÓN

Por último se realiza la comparativa de máximo de llamadas soportadas en cada ancho de banda manteniendo el tiempo de SF en 120 ms. En este caso se incluyen los cuatro modos de máximo funcionamiento del estándar 802.11n según el número de antenas que se utilizan. Las simulaciones se hacen tanto en modo DCF/PCF como en EDCA/HCCA:

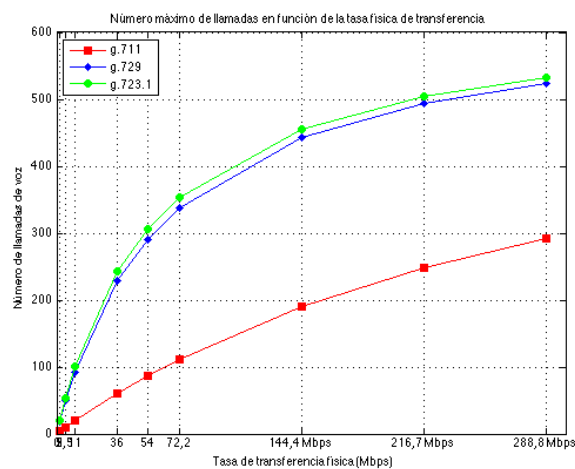


Ilustración 21 - Número máximo de llamadas soportado en DCF/PCF con SF=120ms en función del incremento de la velocidad de transmisión

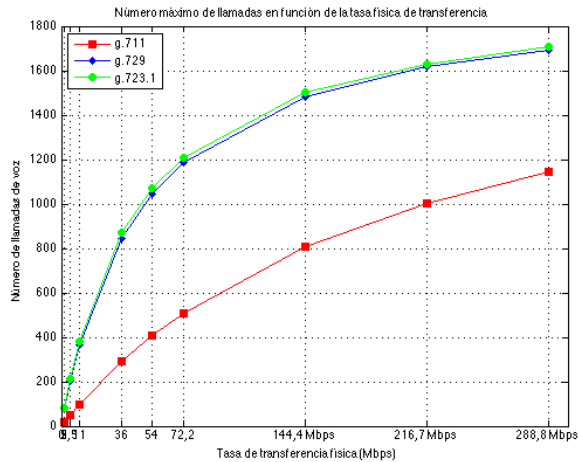


Ilustración 22 - Número máximo de llamadas soportado en EDCA/HCCA con SF=120ms en función del incremento de la velocidad de transmisión

Como en el caso anterior, se ve con el incremento de ancho de banda de la red, se produce un gran incremento en el número de llamadas. En este caso se ve que la gran capacidad que proporciona el protocolo 802.11n permite el poder tener muchas más llamadas que con los protocolos anteriores. Las velocidades máximas de los cuatro modos del 802.11n se han sacado de los índices MCS definidos para este estándar. A modo resumen se ponen dos tablas con los valores numéricos para ver el incremento:

Physical Rate	G.711 (64 Kbps)	G.729 (8 Kbps)	G.723.1 (6,3 Kbps)
2 Mbps	4	19	46
5,5 Mbps	10	51	58
11 Mbps	20	95	107
36 Mbps	61	232	254
54 Mbps	87	295	318
72,2 Mbps	111	338	353
144,4 Mbps	190	442	455
216,7 Mbps	250	493	504
288,8 Mbps	293	523	532

Tabla 19 - Número máximo de llamadas para los codecs estándar en función de la tasa de transmisión y usando el modo de funcionamiento DCF/PCF

Physical Rate	G.711 (64 Kbps)	G.729 (8 Kbps)	G.723.1 (6,3 Kbps)
2 Mbps	19	77	86
5,5 Mbps	51	199	219
11 Mbps	99	360	392
36 Mbps	288	813	861
54 Mbps	400	997	1045
72,2 Mbps	510	1185	1210
144,4 Mbps	807	1481	1501
216,7 Mbps	1004	1616	1631
288,8 Mbps	1143	1693	1705

Tabla 20 - Número máximo de llamadas para los codecs estándar en función de la tasa de transmisión y usando el modo de funcionamiento EDCA/HCCA

A la luz de estos resultados se ve que a medida que crecen las velocidades de la capa física se aumentan mucho las capacidades a la hora de poder transmitir llamadas de VoIP por la red wireless. Incluso con estos números se puede empezar a pensar en usar el códec G.711 en vez de los codecs de compresión

como el G.729 y el G.723.1 ya que el primero es menos sensible a los retardos y mantiene una muy buena calidad de la voz.

De hecho, con estos valores se podría pensar en usar el nuevo códec G.711.1 si se quiere obtener una calidad de voz mucho mayor todavía. Otra posible ampliación podría haber sido intentar medir el número de llamadas con este códec usando el estándar 802.11n.

7.- CONCLUSIONES

Tras las simulaciones realizadas, se puede determinar de manera muy evidente que las mejoras propuestas por el estándar 802.11e son muy buenas, ya que el rendimiento de la red para su uso con llamadas de VoIP mejora considerablemente. El hecho de poder usar los tiempos de CP para poder realizar sondeos y así poder tramitar tráfico, es el hecho que permite esta gran mejora. Además, con los anchos de banda más actuales, se puede plantear el hecho de usar el códec G.711 para realizar estas llamadas.

El códec G.711 tiene una mejor calidad de voz que los otros, y es menos sensible a los retardos del tráfico. Estos hechos unidos al gran número de llamadas que se pueden conseguir usando el estándar 802.11n son los que hacen pensar que es una muy buena alternativa para que la experiencia del usuario realizando llamadas a través de una red *wireless* sea lo más parecida posible a la telefonía convencional.

También hay que destacar que en el mundo de la VoIP, un escenario no se restringe única y exclusivamente a la red *wireless* en la que se encuentra un usuario. Las llamadas hay que cursarlas a través de Internet. Ahí es donde no siempre se puede disponer de un ancho de banda tan grande como el que se plantea para las redes wifi más modernas. Se podrían usar redes por fibra óptica, pero tienen un alto coste todavía. Es por esto que los codecs de compresión como el G.729 y el G.723.1 tienen una alta importancia, ya que el ahorro de ancho de banda es un ahorro económico directo.

Otro factor a la hora de mandar las llamadas por una red pública como Internet, es que las llamadas de VoIP tienen que compartir el medio con otros tipos de tráfico, que van a impedir que se pueda hacer uso de todo el ancho de banda para las llamadas.

Además, se plantea el problema de la seguridad al enviar las llamadas. Alguien que capture el tráfico podría reconstruir el audio enviado a través de RTP y escuchar una conversación. Existen alternativas para este hecho. Se puede cifrar el tráfico SIP para que un posible atacante no pueda ver los extremos de la conversación y así sacar información adicional de la conversación. Este tipo de tráfico se denomina SIP+TLS. Con este protocolo conseguimos cifrar el tráfico SIP entre origen y destino. Realmente es entre origen y PBX y desde ésta hasta el destino. Para la parte de voz se puede usar el protocolo SRTP, el cual es igual que el RTP pero cifrando la parte donde va la voz codificada.

El problema de estas soluciones es que añaden retardos para poder cifrar la información y los anchos de banda que se han calculado para la simulación tampoco son los mismos, ya que los paquetes ocupan más ancho de banda. Aun así, la seguridad en las comunicaciones es un hecho que preocupa mucho, y es preferible poder hacer menos llamadas pero al menos hacerlas con la tranquilidad de que nadie puede hacer escuchas de una manera sencilla.

El otro gran hecho que queda demostrado con esta simulación es que ajustando de manera adecuada los tiempos de la *superframe* y del periodo de CP, podemos conseguir una gran mejora de rendimiento. Esto demuestra que con los mismos recursos se mejora el rendimiento gracias a que el protocolo nos permite variar estos parámetros.

Aún así, en estas simulaciones se ha supuesto que no se tiene pérdida de paquetes ni cómo afecta este hecho al retardo que se produce en el envío de los paquetes. En un escenario real seguramente el número de llamadas simultáneas sea menor debido a estos hechos. A pesar de esto, los números obtenidos hacen pensar también que se podrían diseñar unos dispositivos especializados en VoIP. Estos serían los que traerían ya preparados todos los parámetros para que el rendimiento del número de llamadas sea el máximo posible con las mejores calidades.

Y tal y como se ha comentado, estos dispositivos son los que llevan el distintivo de WMM (*WiFi Multimedia*). Además, la *WiFi Alliance*¹⁶ da las certificaciones para todos aquellos dispositivos que cumplen con el estándar y les dan el sello de WMM. Con estos dispositivos podemos estar seguros que vamos a tener el mejor rendimiento para la transmisión de voz y de vídeo.

Como se ve, la tecnología crece muy deprisa. Aquí sólo se ha llegado hasta el estándar n de la familia 802.11, pero ya existe otro estándar, el ac, que llega hasta velocidades gigabit. Con estos anchos de banda, más el QoS implementado en los dispositivos gracias al 802.11e, se tiene asegurado un gran número de llamadas simultáneas y con buena calidad.

Por otra parte, sólo se han comentado tres de los codecs más habituales, pero también se siguen desarrollando nuevos codecs que mejoren todavía más a estos tres. Uno de ellos es un códec llamado “opus”¹⁷. Parece ser que soporta muchas configuraciones¹⁸ de codificación de audio, pudiendo ser tanto CBR como VBR, y yendo desde los 6 Kbps hasta los 510 kbps. Esto le confieren un gran espectro de funcionalidades y se le empieza a llamar el mejor códec del mundo.

En resumen, el estudio que se ha presentado en el presente TFC da una idea del aprovechamiento que se puede hacer de una red wireless que soporte 802.11e frente a las que no lo soporten para el rendimiento de la VoIP. También se ha visto que aunque se mejore mucho el rendimiento sólo con el QoS que ofrece este estándar de WiFi, también es muy importante la elección del códec que se quiera usar en las comunicaciones, ya que este factor será también determinante para la calidad de las llamadas.

Como se ha visto ya al final de este trabajo, la tecnología no termina de desarrollarse nunca y parece que ésta se encamina hacia la mejora constante tanto de las redes wireless como de crear nuevos codecs con mejores calidades y aprovechamiento de ancho de banda para poder ofrecer servicios de voz sobre redes de bajo rendimiento.

Con lo que se podrá seguir investigando en la línea de este TFC para averiguar si con las nuevas tecnologías se consigue mejorar los resultados aquí presentados.

¹⁶ <http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-voice-programs>

¹⁷ <http://www.opus-codec.org/>

¹⁸ <http://www.sinologic.net/blog/2012-09/ha-nacido-un-nuevo-codec-opus.html#.U5wkTi8Tlul>

8.- MATERIAL DE REFERENCIA

1. <http://www.itu.int/rec/T-REC-G.723.1-200605-I/en>
2. <http://en.wikipedia.org/wiki/G.729>
3. <http://en.wikipedia.org/wiki/G.723.1>
4. <http://en.wikipedia.org/wiki/G.711>
5. http://en.wikipedia.org/wiki/Mean_Opinion_Score
6. http://es.wikipedia.org/wiki/Voz_sobre_Protocolo_de_Internet
7. <http://es.wikipedia.org/wiki/G.711>
8. <http://es.wikipedia.org/wiki/G.729>
9. <http://www.itu.int/rec/T-REC-G.729-201206-I/es>
10. http://en.wikipedia.org/wiki/Voice_over_IP
11. <http://en.wikipedia.org/wiki/H.323>
12. http://en.wikipedia.org/wiki/Real-time_Transport_Protocol
13. http://en.wikipedia.org/wiki/Session_Initiation_Protocol
14. <http://www.voipforo.com/codec/codecs.php>
15. <http://www.voipforo.com/codec/codec-g711--ley.php>
16. http://es.wikipedia.org/wiki/IEEE_802.11
17. http://en.wikipedia.org/wiki/IEEE_802.11e
18. http://en.wikipedia.org/wiki/IEEE_802.11
19. http://en.wikipedia.org/wiki/IEEE_802.11a-1999
20. http://es.wikipedia.org/wiki/IEEE_802.11b
21. http://en.wikipedia.org/wiki/IEEE_802.11b-1999
22. http://en.wikipedia.org/wiki/IEEE_802.11g-2003
23. http://en.wikipedia.org/wiki/List_of_WLAN_channels
24. http://en.wikipedia.org/wiki/Electromagnetic_interference_at_2.4_GHz
25. http://en.wikipedia.org/wiki/IEEE_802.11n-2009
26. http://en.wikipedia.org/wiki/Arbitration_inter-frame_spacing
27. http://es.wikipedia.org/wiki/Funci%C3%B3n_de_coordinaci%C3%B3n_distribuida
28. http://en.wikipedia.org/wiki/DCF_Interframe_Space
29. <http://en.wikipedia.org/wiki/SIFS>
30. http://en.wikipedia.org/wiki/Point_coordination_function
31. http://en.wikipedia.org/wiki/PCF_Interframe_Space
32. M. Veeraraghavan, N. Cocker, and T. Moors. *Support of voice services in IEEE 802.11 Wireless LANs*. In Proceedings of IEEE INFOCOM 2001, Alaska, April 2001.
33. IEEE Std 802.11-2012, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, March 2012
34. IEEE Std 802.11a-1999(R2003), Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band*, June 2003
35. IEEE Std 802.11b-1999(R2003), Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, June 2003
36. IEEE Std 802.11e-2005, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, November 2005
37. IEEE Std 802.11g-2003, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*, June 2003
38. IEEE Std 802.11n-2009, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 5: Enhancements for Higher Throughput*, October 2009
39. <http://www.itu.int/rec/T-REC-G.107-201112-S/en>
40. http://www.sageinst.com/downloads/960B/EModel_wp.pdf

41. <https://www.itu.int/rec/T-REC-G.113-200711-I/en>
42. <http://www.itu.int/ITU-T/studygroups/com12/emodelv1/index.htm>
43. http://www.hamilton.ie/dwmalone/assem_et_al.pdf
44. http://netserver.ics.forth.gr/wiki/images/3/3a/VOIP_MOS_models.pdf
45. <http://www.voipforo.com/IAX/IAXvsSIP.php>
46. <https://sites.google.com/site/3cuelectronica/home/voip/sip>

ANEXO I - CÓDIGO DE MATLAB DE LA SIMULACIÓN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% TFC de César Solera Calvo
%
% Fecha: 28/05/14
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function simulacion_80211e()
%Función principal de la simulación

%Cálculo del valor de MOS en función del retardo
%R0 = 93.2;
R0 = 94.2; %Valor de la constante R0

g711_Ie = 0; %Valor de Ie para g711
g729_Ie = 11; %Valor de Ie para g729
g723_Ie = 19; %Valor de Ie para g723.1

d = 0:0.1:500; %Variable para medir del retardo

y1 = MOS(Rfactor(R0,d,g711_Ie));
y2 = MOS(Rfactor(R0,d,g729_Ie));
y3 = MOS(Rfactor(R0,d,g723_Ie));
y4 = 3.6;

% Create figure
plot(d,y1,'r',d,y2,'b',d,y3,'g',d,y4,'k');
xlabel('Retardo(ms)');
ylabel('MOS');
title('MOS de codecs estandar en función del retardo');
legend('g.711','g.729','g.723.1');
grid on

%Fin del cálculo de MOS

%Valores constantes para toda la simulación
g711_CBR = 64000; %Valor de CBR para g711
g729_CBR = 8000; %Valor de CBR para g729
g723_CBR = 6300; %Valor de CBR para g723.1

g711_Tvoice = 0.02; %Valor de tiempo de paquetización para g711
g729_Tvoice = 0.025; %Valor de tiempo de paquetización para g729
g723_Tvoice = 0.0375; %Valor de tiempo de paquetización para g723.1

%MAC80211 = 2e6;
%a80211 = 5.5e6;
b80211 = 11000000; %Valor de la velocidad de transmisión para 802.11b
%g80211 = 54e6;

B = 320; %Beacon frame size
H = 456; %Protocol header overhead
P = 192; %Physical layer header
Tb = 0.01; %Beacon interval
SIFS = 0.000028; %SIFS time interval
CFend = 192; %CFP end frame
```

```
%Cálculo del Número máximo de llamadas en DCF/PCF en función de Tsf
%variable

Tsf0 = 0.030:0.010:0.120; %Tiempo de la superframe
Tcfp0 = Tsf0 ./ 2.4;
Tcp0 = Tsf0 - Tcfp0;

y20 = Ntotal_DCF(g711_CBR,g711_Tvoice,Tcfp0,H,P,b80211,SIFS,Tcp0,B,Tsf0,Tb,CFend);
y21 = Ntotal_DCF(g729_CBR,g729_Tvoice,Tcfp0,H,P,b80211,SIFS,Tcp0,B,Tsf0,Tb,CFend);
y22 = Ntotal_DCF(g723_CBR,g723_Tvoice,Tcfp0,H,P,b80211,SIFS,Tcp0,B,Tsf0,Tb,CFend);

% Create figure
figure2 = figure;

% Create axes
axes2 = axes('Parent',figure2,...
    'XTickLabel',{'30','40','50','60','70','80','90','100','110','120'},...
    'XTick',[0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12]);
box(axes2,'on');
grid(axes2,'on');
hold(axes2,'all');

% Create multiple lines using matrix input to plot
plot2 = plot(Tsf0,y20,Tsf0,y21,Tsf0,y22,'Parent',axes2);
set(plot2(1),'MarkerFaceColor',[1 0 0],'Marker','none','Color',[1 0 0],...
    'DisplayName','g.711');
set(plot2(2),'MarkerFaceColor',[0 0 1],'Marker','none','Color',[0 0 1],...
    'DisplayName','g.729');
set(plot2(3),'MarkerFaceColor',[0 1 0],'Marker','none','Color',[0 1 0],...
    'DisplayName','g.723.1');

% Create xlabel
xlabel('SF(ms)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas soportadas en modo DCF/PCF');

% Create legend
legend2 = legend(axes2,'show');
set(legend2,'Location','NorthWest');

%Fin cálculo llamadas en DCF/PCF en función de Tsf variable

%Cálculo del Número máximo de llamadas en EDCA/HCCA en función de Tsf
%variable

Tsf = 0.030:0.010:0.120; %Tiempo de la superframe
Tcfp = Tsf ./ 3;
Tcp = Tsf - Tcfp;

y5 = Ntotal(g711_CBR,g711_Tvoice,Tcfp,H,P,b80211,SIFS,Tcp,B,Tsf,Tb,CFend);
y6 = Ntotal(g729_CBR,g729_Tvoice,Tcfp,H,P,b80211,SIFS,Tcp,B,Tsf,Tb,CFend);
y7 = Ntotal(g723_CBR,g723_Tvoice,Tcfp,H,P,b80211,SIFS,Tcp,B,Tsf,Tb,CFend);
```

```
% Create figure
figure3 = figure;

% Create axes
axes3 = axes('Parent',figure3,...
    'XTickLabel',{'30','40','50','60','70','80','90','100','110','120'},...
    'XTick',[0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12]);
box(axes3,'on');
grid(axes3,'on');
hold(axes3,'all');

% Create multiple lines using matrix input to plot
plot3 = plot(Tsf,y5,Tsf,y6,Tsf,y7,'Parent',axes3);
set(plot3(1),'MarkerFaceColor',[1 0 0],'Marker','none','Color',[1 0 0],...
    'DisplayName','g.711');
set(plot3(2),'MarkerFaceColor',[0 0 1],'Marker','none','Color',[0 0 1],...
    'DisplayName','g.729');
set(plot3(3),'MarkerFaceColor',[0 1 0],'Marker','none','Color',[0 1 0],...
    'DisplayName','g.723.1');

% Create xlabel
xlabel('SF(ms)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas soportadas en modo EDCA/HCCA');

% Create legend
legend3 = legend(axes3,'show');
set(legend3,'Location','NorthWest');

%Fin cálculo llamadas en EDCA/HCCA en función de Tsf variable

%Cálculo del Número máximo de llamadas en EDCA/HCCA en función de Tcp
%variable

Tcp2 = 0.03:0.01:0.12;
Tsf2 = 0.12;
Tcfp2 = Tsf2 - Tcp2;

y8 = Ntotal(g711_CBR,g711_Tvoice,Tcfp2,H,P,b80211,SIFS,Tcp2,B,Tsf2,Tb,CFend);
y9 = Ntotal(g729_CBR,g729_Tvoice,Tcfp2,H,P,b80211,SIFS,Tcp2,B,Tsf2,Tb,CFend);
y10 = Ntotal(g723_CBR,g723_Tvoice,Tcfp2,H,P,b80211,SIFS,Tcp2,B,Tsf2,Tb,CFend);

% Create figure
figure4 = figure;

% Create axes
axes4 = axes('Parent',figure4,...
    'XTickLabel',{'30','40','50','60','70','80','90','100','110','120'},...
    'XTick',[0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12]);
box(axes4,'on');
grid(axes4,'on');
hold(axes4,'all');

% Create multiple lines using matrix input to plot
```

```
plot4 = plot(Tcp2,y8,Tcp2,y9,Tcp2,y10,'Parent',axes4);
set(plot4(1),'MarkerFaceColor',[1 0 0],'Marker','none','Color',[1 0 0],...
'DisplayName','g.711');
set(plot4(2),'MarkerFaceColor',[0 0 1],'Marker','none','Color',[0 0 1],...
'DisplayName','g.729');
set(plot4(3),'MarkerFaceColor',[0 1 0],'Marker','none','Color',[0 1 0],...
'DisplayName','g.723.1');

% Create xlabel
xlabel('CP(ms)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas soportadas en modo EDCA/HCCA');

% Create legend
legend4 = legend(axes4,'show');
set(legend4,'Location','NorthEast');

%Fin cálculo llamadas en EDCA/HCCA en función de Tcp variable

%Cálculo del número máximo de llamadas en función de la tasa de
%transferencia usada en modo EDCA/HCCA
Tcp3 = 0.077;
Tsf3 = 0.12;
Tcfp3 = Tsf3 - Tcp3;
Tasa = [2e6 5.5e6 11e6 36e6 54e6]; %Velocidades sobre las que se calcula

y11 = Ntotal(g711_CBR,g711_Tvoice,Tcfp3,H,P,Tasa,SIFS,Tcp3,B,Tsf3,Tb,CFend);
y12 = Ntotal(g729_CBR,g729_Tvoice,Tcfp3,H,P,Tasa,SIFS,Tcp3,B,Tsf3,Tb,CFend);
y13 = Ntotal(g723_CBR,g723_Tvoice,Tcfp3,H,P,Tasa,SIFS,Tcp3,B,Tsf3,Tb,CFend);

% Create figure
figure5 = figure;

% Create axes
axes5 = axes('Parent',figure5,...
'XTickLabel',{'0','2','5.5','11 Mbps','36 Mbps','54 Mbps'},...
'XTick',[0 2000000 5500000 11000000 36000000 54000000]);
box(axes5,'on');
grid(axes5,'on');
hold(axes5,'all');

% Create multiple lines using matrix input to plot
plot5 = plot(Tasa,y11,Tasa,y12,Tasa,y13,'Parent',axes5);
set(plot5(1),'MarkerFaceColor',[1 0 0],'Marker','square','Color',[1 0 0],...
'DisplayName','g.711');
set(plot5(2),'MarkerFaceColor',[0 0 1],'Marker','diamond','Color',[0 0 1],...
'DisplayName','g.729');
set(plot5(3),'MarkerFaceColor',[0 1 0],'Marker','o','Color',[0 1 0],...
'DisplayName','g.723.1');

% Create xlabel
xlabel('Tasa de transferencia física (Mbps)');

% Create ylabel
ylabel('Número de llamadas de voz');
```



```
% Create title
title('Número máximo de llamadas en función de la tasa física de transferencia');

% Create legend
legend5 = legend(axes5, 'show');
set(legend5, 'Location', 'NorthWest');

%Fin para el cálculo de llamadas en modo EDCA/HCCA en función de la tasa de
%transferencia

%Cálculo del número máximo de llamadas en función de la tasa de
%transferencia usada en modo DCF/PCF
Tcp4 = 0.07;
Tsf4 = 0.12;
Tcfp4 = Tsf4 - Tcp4;
Tasa2 = [2e6 5.5e6 11e6 36e6 54e6]; %Velocidades sobre las que se calcula

y14 = Ntotal_DCF(g711_CBR,g711_Tvoice,Tcfp4,H,P,Tasa2,SIFS,Tcp4,B,Tsf4,Tb,CFend);
y15 = Ntotal_DCF(g729_CBR,g729_Tvoice,Tcfp4,H,P,Tasa2,SIFS,Tcp4,B,Tsf4,Tb,CFend);
y16 = Ntotal_DCF(g723_CBR,g723_Tvoice,Tcfp4,H,P,Tasa2,SIFS,Tcp4,B,Tsf4,Tb,CFend);

% Create figure
figure6 = figure;

% Create axes
axes6 = axes('Parent',figure6,...
    'XTickLabel',{ '0', '2', '5,5', '11 Mbps', '36 Mbps', '54 Mbps'},...
    'XTick',[0 2000000 5500000 11000000 36000000 54000000]);
box(axes6, 'on');
grid(axes6, 'on');
hold(axes6, 'all');

% Create multiple lines using matrix input to plot
plot6 = plot(Tasa2,y14,Tasa2,y15,Tasa2,y16, 'Parent', axes6);
set(plot6(1), 'MarkerFaceColor',[1 0 0], 'Marker', 'square', 'Color',[1 0 0],...
    'DisplayName', 'g.711');
set(plot6(2), 'MarkerFaceColor',[0 0 1], 'Marker', 'diamond', 'Color',[0 0 1],...
    'DisplayName', 'g.729');
set(plot6(3), 'MarkerFaceColor',[0 1 0], 'Marker', 'o', 'Color',[0 1 0],...
    'DisplayName', 'g.723.1');

% Create xlabel
xlabel('Tasa de transferencia física (Mbps)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas en función de la tasa física de transferencia');

% Create legend
legend6 = legend(axes6, 'show');
set(legend6, 'Location', 'NorthWest');

%Fin para el cálculo de llamadas en modo DCF/PCF en función de la tasa de
%transferencia
```

end

```
function Id = Id(d)
%Función para calcular el retardo
    if d < 400
    then
        Id = 0.024.*d+0.11.*(d-177.3).*heaviside(d-177.3);
    else
        Id = -2.468e-14.*d.^6+5.062e-11.*d.^5-3.903e-8.*d.^4+1.344e-5.*d.^3-
0.001802.*d.^2+0.103.*d-0.1698;
    end
end
```

```
function Rfactor = Rfactor(R0,d,Ie)
%Función para calcular el factor R
    Rfactor = R0-Id(d)-Ie;
end
```

```
function MOS = MOS(R)
%Función para calcular el MOS a partir del factor R
    MOS = 1+0.035.*R+7e-6.*R.*(R-60).*(100-R);
end
```

```
function Tpoll_cfp = Tpoll_cfp(CBR,Tvoice,Tcfp,H,P,R,SIFS)
%Función para calcular el tiempo de sondeo durante CFP en modo
%EDCA/HCCA
    aux1 = Tvoice + Tcfp;
    aux2 = CBR .* aux1;
    aux3 = (aux2 + H + P) .* 2;
    aux4 = aux3 ./ R;
    Tpoll_cfp = aux4 + (2 .* SIFS);
end
```

```
function Tpoll_dcf = Tpoll_dcf(CBR,Tvoice,Tsf,H,P,R,SIFS)
%Función para calcular el tiempo de sondeo durante CFP en
%el modo DCF/PCF
    aux1 = Tvoice + Tsf;
    aux2 = CBR .* aux1;
    aux3 = (aux2 + H + P) .* 2;
    aux4 = aux3 ./ R;
    Tpoll_dcf = aux4 + (2 .* SIFS);
end
```

```
function Tpoll_cp = Tpoll_cp(CBR,Tvoice,Tcp,H,P,R,SIFS)
%Función para calcular el tiempo de sondeo durante CP
    aux1 = Tvoice + Tcp;
    aux2 = CBR .* aux1;
    aux3 = (aux2 + H + P) .* 2;
    aux4 = aux3 ./ R;
    Tpoll_cp = aux4 + (2 .* SIFS);
end
```

```
function Toverhead = Toverhead(B,P,R,SIFS,Tsf,Tcp,Tb,CFend)
%Función para calcular el tiempo de overhead asociado al CFP
    aux1 = ((B+P)./R) + SIFS;
    aux2 = (Tsf - Tcp)./Tb;
    aux3 = (CFend + P)./R;
    Toverhead = aux1 .* aux2 + aux3;
end
```

```
function Npoll_cfp = Npoll_cfp(Tcfp,Toverhead,Tpoll_cfp,Tcp)
%Función para calcular el total de llamadas durante el tiempo de CFP
    %Npoll_cfp = (Tcfp - Toverhead) ./ Tpoll_cfp; %Función original del
    %autor
    Npoll_cfp = ((Tcfp - Toverhead) + 0.03.*Tcp)./ Tpoll_cfp; %Modificación para
conseguir los mismos resultados
end

function Npoll_dcf = Npoll_dcf(Tcfp,Toverhead,Tpoll_cfp,Tcp)
%Función para calcular el total de llamadas en modo DCF/PCF
    %Npoll_dcf = (Tcfp - Toverhead) ./ Tpoll_cfp; %Función original del
    %autor
    Npoll_dcf = ((Tcfp - Toverhead) - 0.2.*Tcp)./ Tpoll_cfp; %Modificación para
conseguir los mismos resultados
end

function Npoll_cp = Npoll_cp(Tcp,Tpoll_cp)
%Función para calcular el total de llamadas durante el tiempo de CP
    %Npoll_cp = Tcp ./ Tpoll_cp; Función original del autor
    Npoll_cp = (Tcp - 0.15.*Tcp)./ Tpoll_cp; %Modificación para conseguir los mismos
resultados
end

function Ntotal = Ntotal(CBR,Tvoice,Tcfp,H,P,R,SIFS,Tcp,B,Tsf,Tb,CFend)
%Función para calcular el número total de llamadas en modo EDCA/HCCA
    aux1 = Tpoll_cfp(CBR,Tvoice,Tcfp,H,P,R,SIFS);
    aux2 = Tpoll_cp(CBR,Tvoice,Tcp,H,P,R,SIFS);
    aux3 = Toverhead(B,P,R,SIFS,Tsf,Tcp,Tb,CFend);
    Ntotal = Npoll_cfp(Tcfp,aux3,aux1,Tcp) + Npoll_cp(Tcp,aux2);
end

function Ntotal_DCF = Ntotal_DCF(CBR,Tvoice,Tcfp,H,P,R,SIFS,Tcp,B,Tsf,Tb,CFend)
%Función para calcular el número total de llamadas en modo DCF/PCF
    aux1 = Tpoll_dcf(CBR,Tvoice,Tsf,H,P,R,SIFS);
    aux3 = Toverhead(B,P,R,SIFS,Tsf,Tcp,Tb,CFend);
    Ntotal_DCF = Npoll_dcf(Tcfp,aux3,aux1,Tcp);
end
```

ANEXO II - CÓDIGO DE MATLAB DE LA SIMULACIÓN AMPLIADA

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% TFC de César Solera Calvo
%
% Fecha: 28/05/14
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function simulacion_80211e_ampliacion()
%Función principal de la simulación

%Cálculo del valor de MOS en función del retardo
%R0 = 93.2;
R0 = 94.2; %Valor de la constante R0

g711_Ie = 0; %Valor de Ie para g711
g729_Ie = 11; %Valor de Ie para g729
g723_Ie = 19; %Valor de Ie para g723.1

d = 0:0.1:500; %Variable para medir del retardo

y1 = MOS(Rfactor(R0,d,g711_Ie));
y2 = MOS(Rfactor(R0,d,g729_Ie));
y3 = MOS(Rfactor(R0,d,g723_Ie));
y4 = 3.6;

% Create figure
plot(d,y1,'r',d,y2,'b',d,y3,'g',d,y4,'k');
xlabel('Retardo(ms)');
ylabel('MOS');
title('MOS de codecs estandar en función del retardo');
legend('g.711','g.729','g.723.1');
grid on

%Fin del cálculo de MOS

%Valores constantes para toda la simulación
g711_CBR = 64000; %Valor de CBR para g711
g729_CBR = 8000; %Valor de CBR para g729
g723_CBR = 6300; %Valor de CBR para g723.1

g711_Tvoice = 0.02; %Valor de tiempo de paquetización para g711
g729_Tvoice = 0.025; %Valor de tiempo de paquetización para g729
g723_Tvoice = 0.0375; %Valor de tiempo de paquetización para g723.1

%MAC80211 = 2e6;
%a80211 = 5.5e6;
%b80211 = 11000000; %Valor de la velocidad de transmisión para 802.11b
g80211 = 54e6; %Valor de la velocidad de transmisión para 802.11g

B = 320; %Beacon frame size
H = 456; %Protocol header overhead
P = 192; %Physical layer header
Tb = 0.01; %Beacon interval
SIFS = 0.000028; %SIFS time interval
CFend = 192; %CFP end frame
```

```
%Cálculo del Número máximo de llamadas en DCF/PCF en función de Tsf
%variable

Tsf0 = 0.030:0.010:0.120; %Tiempo de la superframe
Tcfp0 = Tsf0 ./ 2.4;
Tcp0 = Tsf0 - Tcfp0;

y20 = Ntotal_DCF(g711_CBR,g711_Tvoice,Tcfp0,H,P,g80211,SIFS,Tcp0,B,Tsf0,Tb,CFend);
y21 = Ntotal_DCF(g729_CBR,g729_Tvoice,Tcfp0,H,P,g80211,SIFS,Tcp0,B,Tsf0,Tb,CFend);
y22 = Ntotal_DCF(g723_CBR,g723_Tvoice,Tcfp0,H,P,g80211,SIFS,Tcp0,B,Tsf0,Tb,CFend);

% Create figure
figure2 = figure;

% Create axes
axes2 = axes('Parent',figure2,...
    'XTickLabel',{'30','40','50','60','70','80','90','100','110','120'},...
    'XTick',[0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12]);
box(axes2,'on');
grid(axes2,'on');
hold(axes2,'all');

% Create multiple lines using matrix input to plot
plot2 = plot(Tsf0,y20,Tsf0,y21,Tsf0,y22,'Parent',axes2);
set(plot2(1),'MarkerFaceColor',[1 0 0],'Marker','none','Color',[1 0 0],...
    'DisplayName','g.711');
set(plot2(2),'MarkerFaceColor',[0 0 1],'Marker','none','Color',[0 0 1],...
    'DisplayName','g.729');
set(plot2(3),'MarkerFaceColor',[0 1 0],'Marker','none','Color',[0 1 0],...
    'DisplayName','g.723.1');

% Create xlabel
xlabel('SF(ms)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas soportadas en modo DCF/PCF');

% Create legend
legend2 = legend(axes2,'show');
set(legend2,'Location','NorthWest');

%Fin cálculo llamadas en DCF/PCF en función de Tsf variable

%Cálculo del Número máximo de llamadas en EDCA/HCCA en función de Tsf
%variable

Tsf = 0.030:0.010:0.120; %Tiempo de la superframe
Tcfp = Tsf ./ 3;
Tcp = Tsf - Tcfp;

y5 = Ntotal(g711_CBR,g711_Tvoice,Tcfp,H,P,g80211,SIFS,Tcp,B,Tsf,Tb,CFend);
y6 = Ntotal(g729_CBR,g729_Tvoice,Tcfp,H,P,g80211,SIFS,Tcp,B,Tsf,Tb,CFend);
y7 = Ntotal(g723_CBR,g723_Tvoice,Tcfp,H,P,g80211,SIFS,Tcp,B,Tsf,Tb,CFend);
```

```
% Create figure
figure3 = figure;

% Create axes
axes3 = axes('Parent',figure3,...
    'XTickLabel',{ '30', '40', '50', '60', '70', '80', '90', '100', '110', '120'},...
    'XTick',[0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12]);
box(axes3,'on');
grid(axes3,'on');
hold(axes3,'all');

% Create multiple lines using matrix input to plot
plot3 = plot(Tsf,y5,Tsf,y6,Tsf,y7,'Parent',axes3);
set(plot3(1),'MarkerFaceColor',[1 0 0],'Marker','none','Color',[1 0 0],...
    'DisplayName','g.711');
set(plot3(2),'MarkerFaceColor',[0 0 1],'Marker','none','Color',[0 0 1],...
    'DisplayName','g.729');
set(plot3(3),'MarkerFaceColor',[0 1 0],'Marker','none','Color',[0 1 0],...
    'DisplayName','g.723.1');

% Create xlabel
xlabel('SF(ms)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas soportadas en modo EDCA/HCCA');

% Create legend
legend3 = legend(axes3,'show');
set(legend3,'Location','NorthWest');

%Fin cálculo llamadas en EDCA/HCCA en función de Tsf variable

%Cálculo del Número máximo de llamadas en EDCA/HCCA en función de Tcp
%variable

Tcp2 = 0.03:0.01:0.12;
Tsf2 = 0.12;
Tcfp2 = Tsf2 - Tcp2;

y8 = Ntotal(g711_CBR,g711_Tvoice,Tcfp2,H,P,g80211,SIFS,Tcp2,B,Tsf2,Tb,CFend);
y9 = Ntotal(g729_CBR,g729_Tvoice,Tcfp2,H,P,g80211,SIFS,Tcp2,B,Tsf2,Tb,CFend);
y10 = Ntotal(g723_CBR,g723_Tvoice,Tcfp2,H,P,g80211,SIFS,Tcp2,B,Tsf2,Tb,CFend);

% Create figure
figure4 = figure;

% Create axes
axes4 = axes('Parent',figure4,...
    'XTickLabel',{ '30', '40', '50', '60', '70', '80', '90', '100', '110', '120'},...
    'XTick',[0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12]);
box(axes4,'on');
grid(axes4,'on');
hold(axes4,'all');

% Create multiple lines using matrix input to plot
```

```
plot4 = plot(Tcp2,y8,Tcp2,y9,Tcp2,y10,'Parent',axes4);
set(plot4(1),'MarkerFaceColor',[1 0 0],'Marker','none','Color',[1 0 0],...
'DisplayName','g.711');
set(plot4(2),'MarkerFaceColor',[0 0 1],'Marker','none','Color',[0 0 1],...
'DisplayName','g.729');
set(plot4(3),'MarkerFaceColor',[0 1 0],'Marker','none','Color',[0 1 0],...
'DisplayName','g.723.1');

% Create xlabel
xlabel('CP(ms)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas soportadas en modo EDCA/HCCA');

% Create legend
legend4 = legend(axes4,'show');
set(legend4,'Location','NorthEast');

%Fin cálculo llamadas en EDCA/HCCA en función de Tcp variable

%Cálculo del número máximo de llamadas en función de la tasa de
%transferencia usada en modo EDCA/HCCA
Tcp3 = 0.077;
Tsf3 = 0.12;
Tcfp3 = Tsf3 - Tcp3;
Tasa = [2e6 5.5e6 11e6 36e6 54e6 72.2e6 144.4e6 216.7e6 288.8e6]; %Velocidades sobre
las que se calcula

y11 = Ntotal(g711_CBR,g711_Tvoice,Tcfp3,H,P,Tasa,SIFS,Tcp3,B,Tsf3,Tb,CFend);
y12 = Ntotal(g729_CBR,g729_Tvoice,Tcfp3,H,P,Tasa,SIFS,Tcp3,B,Tsf3,Tb,CFend);
y13 = Ntotal(g723_CBR,g723_Tvoice,Tcfp3,H,P,Tasa,SIFS,Tcp3,B,Tsf3,Tb,CFend);

% Create figure
figure5 = figure;

% Create axes
axes5 = axes('Parent',figure5,...
'XTickLabel',{'0','2','5,5','11','36','54','72,2','144,4 Mbps','216,7 Mbps','288,8
Mbps'},...
'XTick',[0 2000000 5500000 11000000 36000000 54000000 72200000 144400000 216700000
288800000]);
box(axes5,'on');
grid(axes5,'on');
hold(axes5,'all');

% Create multiple lines using matrix input to plot
plot5 = plot(Tasa,y11,Tasa,y12,Tasa,y13,'Parent',axes5);
set(plot5(1),'MarkerFaceColor',[1 0 0],'Marker','square','Color',[1 0 0],...
'DisplayName','g.711');
set(plot5(2),'MarkerFaceColor',[0 0 1],'Marker','diamond','Color',[0 0 1],...
'DisplayName','g.729');
set(plot5(3),'MarkerFaceColor',[0 1 0],'Marker','o','Color',[0 1 0],...
'DisplayName','g.723.1');

% Create xlabel
xlabel('Tasa de transferencia física (Mbps)');
```

```
% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas en función de la tasa física de transferencia');

% Create legend
legend5 = legend(axes5, 'show');
set(legend5, 'Location', 'NorthWest');

%Fin para el cálculo de llamadas en modo EDCA/HCCA en función de la tasa de
%transferencia

%Cálculo del número máximo de llamadas en función de la tasa de
%transferencia usada en modo DCF/PCF
Tcp4 = 0.07;
Tsf4 = 0.12;
Tcfp4 = Tsf4 - Tcp4;
Tasa2 = [2e6 5.5e6 11e6 36e6 54e6 72.2e6 144.4e6 216.7e6 288.8e6]; %Velocidades sobre
las que se calcula

y14 = Ntotal_DCF(g711_CBR,g711_Tvoice,Tcfp4,H,P,Tasa2,SIFS,Tcp4,B,Tsf4,Tb,CFend);
y15 = Ntotal_DCF(g729_CBR,g729_Tvoice,Tcfp4,H,P,Tasa2,SIFS,Tcp4,B,Tsf4,Tb,CFend);
y16 = Ntotal_DCF(g723_CBR,g723_Tvoice,Tcfp4,H,P,Tasa2,SIFS,Tcp4,B,Tsf4,Tb,CFend);

% Create figure
figure6 = figure;

% Create axes
axes6 = axes('Parent',figure6,...
    'XTickLabel',{ '0', '2', '5,5', '11', '36', '54', '72,2', '144,4 Mbps', '216,7 Mbps', '288,8
Mbps'},...
    'XTick',[0 2000000 5500000 11000000 36000000 54000000 72200000 144400000 216700000
288800000]);
box(axes6, 'on');
grid(axes6, 'on');
hold(axes6, 'all');

% Create multiple lines using matrix input to plot
plot6 = plot(Tasa2,y14,Tasa2,y15,Tasa2,y16, 'Parent', axes6);
set(plot6(1), 'MarkerFaceColor',[1 0 0], 'Marker', 'square', 'Color',[1 0 0],...
    'DisplayName', 'g.711');
set(plot6(2), 'MarkerFaceColor',[0 0 1], 'Marker', 'diamond', 'Color',[0 0 1],...
    'DisplayName', 'g.729');
set(plot6(3), 'MarkerFaceColor',[0 1 0], 'Marker', 'o', 'Color',[0 1 0],...
    'DisplayName', 'g.723.1');

% Create xlabel
xlabel('Tasa de transferencia física (Mbps)');

% Create ylabel
ylabel('Número de llamadas de voz');

% Create title
title('Número máximo de llamadas en función de la tasa física de transferencia');

% Create legend
```



```
legend6 = legend(axes6, 'show');
set(legend6, 'Location', 'NorthWest');

%Fin para el cálculo de llamadas en modo DCF/PCF en función de la tasa de
%transferencia

end

function Id = Id(d)
%Función para calcular el retardo
    if d < 400
    then
        Id = 0.024.*d+0.11.*(d-177.3).*heaviside(d-177.3);
    else
        Id = -2.468e-14.*d.^6+5.062e-11.*d.^5-3.903e-8.*d.^4+1.344e-5.*d.^3-
0.001802.*d.^2+0.103.*d-0.1698;
    end
end

function Rfactor = Rfactor(R0,d,Ie)
%Función para calcular el factor R
    Rfactor = R0-Id(d)-Ie;
end

function MOS = MOS(R)
%Función para calcular el MOS a partir del factor R
    MOS = 1+0.035.*R+7e-6.*R.*(R-60).*(100-R);
end

function Tpoll_cfp = Tpoll_cfp(CBR,Tvoice,Tcfp,H,P,R,SIFS)
%Función para calcular el tiempo de sondeo durante el CFP
    aux1 = Tvoice + Tcfp;
    aux2 = CBR .* aux1;
    aux3 = (aux2 + H + P) .* 2;
    aux4 = aux3 ./ R;
    Tpoll_cfp = aux4 + (2 .* SIFS);
end

function Tpoll_dcf = Tpoll_dcf(CBR,Tvoice,Tsf,H,P,R,SIFS)
%Función para calcular el tiempo de sondeo durante el CFP en el modo DCF/PCF
    aux1 = Tvoice + Tsfc;
    aux2 = CBR .* aux1;
    aux3 = (aux2 + H + P) .* 2;
    aux4 = aux3 ./ R;
    Tpoll_dcf = aux4 + (2 .* SIFS);
end

function Tpoll_cp = Tpoll_cp(CBR,Tvoice,Tcp,H,P,R,SIFS)
%Función para calcular el tiempo de sondeo durante el CP
    aux1 = Tvoice + Tcp;
    aux2 = CBR .* aux1;
    aux3 = (aux2 + H + P) .* 2;
    aux4 = aux3 ./ R;
    Tpoll_cp = aux4 + (2 .* SIFS);
end

function Toverhead = Toverhead(B,P,R,SIFS,Tsf,Tcp,Tb,CFend)
%Función para calcular el tiempo de overhead asociado al CFP
    aux1 = ((B+P)./R) + SIFS;
```

```
aux2 = (Tsf - Tcfp)./Tb;
aux3 = (CFend + P)./R;
Toverhead = aux1 .* aux2 + aux3;
end

function Npoll_cfp = Npoll_cfp(Tcfp,Toverhead,Tpoll_cfp,Tcp)
%Función para calcular el total de llamadas durante el tiempo de CFP
    %Npoll_cfp = (Tcfp - Toverhead) ./ Tpoll_cfp; %Función original del
    %autor
    Npoll_cfp = ((Tcfp - Toverhead) + 0.03.*Tcp)./ Tpoll_cfp; %Modificación para
conseguir los mismos resultados
end

function Npoll_dcf = Npoll_dcf(Tcfp,Toverhead,Tpoll_cfp,Tcp)
%Función para calcular el total de llamadas en modo DCF/PCF
    %Npoll_dcf = (Tcfp - Toverhead) ./ Tpoll_cfp; %Función original del
    %autor
    Npoll_dcf = ((Tcfp - Toverhead) - 0.2.*Tcp)./ Tpoll_cfp; %Modificación para
conseguir los mismos resultados
end

function Npoll_cp = Npoll_cp(Tcp,Tpoll_cp)
%Función para calcular el total de llamadas durante el tiempo de CP
    %Npoll_cp = Tcp ./ Tpoll_cp; %Función original del autor
    Npoll_cp = (Tcp - 0.15.*Tcp)./ Tpoll_cp; %Modificación para conseguir los mismos
resultados
end

function Ntotal = Ntotal(CBR,Tvoice,Tcfp,H,P,R,SIFS,Tcp,B,Tsf,Tb,CFend)
%Función para calcular el número total de llamadas en modo EDCA/HCCA
    aux1 = Tpoll_cfp(CBR,Tvoice,Tcfp,H,P,R,SIFS);
    aux2 = Tpoll_cp(CBR,Tvoice,Tcp,H,P,R,SIFS);
    aux3 = Toverhead(B,P,R,SIFS,Tsf,Tcp,Tb,CFend);
    Ntotal = Npoll_cfp(Tcfp,aux3,aux1,Tcp) + Npoll_cp(Tcp,aux2);
end

function Ntotal_DCF = Ntotal_DCF(CBR,Tvoice,Tcfp,H,P,R,SIFS,Tcp,B,Tsf,Tb,CFend)
%Función para calcular el número total de llamadas en modo DCF/PCF
    aux1 = Tpoll_dcf(CBR,Tvoice,Tsf,H,P,R,SIFS);
    aux3 = Toverhead(B,P,R,SIFS,Tsf,Tcp,Tb,CFend);
    Ntotal_DCF = Npoll_dcf(Tcfp,aux3,aux1,Tcp);
end
```