

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



**EVALUACIÓN DE ARQUITECTURAS Y
PROTOCOLOS EN SERVICIOS DE STREAMING
UNICAST Y MULTICAST SOBRE LTE**

TRABAJO FIN DE MÁSTER

Carlos M. Lentisco Sánchez

2014

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**EVALUACIÓN DE ARQUITECTURAS Y
PROTOCOLOS EN SERVICIOS DE STREAMING
UNICAST Y MULTICAST SOBRE LTE**

Autor
Carlos M. Lentisco Sánchez

Director
Luis Bellido Triana

Departamento de Ingeniería de Sistemas Telemáticos

2014

Agradecimientos

En primer lugar, quiero dar las gracias a Encarna y Luis por todo lo que han hecho por mí desde hace ya un año. No solo por la confianza que depositaron en mí desde el día cero y por darme la oportunidad de realizar un trabajo apasionante, sino por su enseñanza incondicional, por su esfuerzo y dedicación para ayudarme a progresar. Este trabajo no habría sido posible sin ellos.

También quiero dar las gracias a todos mis compañeros de trabajo en el Departamento de Ingeniería de Sistemas Telemáticos: A todos los profesores, todos me han ayudado a progresar y de todos y cada uno de ellos he aprendido a lo largo de este año. A los investigadores, por su trabajo y buena disposición. Finalmente, y como no, a mis compañeros de despacho Joaquín y Mario, por ayudarme, aguantarme y hacer que el día a día sea fácil, ameno y divertido.

y como siempre gracias, muchas gracias...

A mi padre que está en el cielo,

a mi madre,

a mi hermano,

a mi único y gran amor Marijose,

a mi familia y amigos,

La vida es aquello que te va sucediendo mientras te empeñas en hacer otros planes

John Lennon

Resumen

Actualmente, los servicios de video streaming sobre redes móviles generan interés en la sociedad de la información. Ésta demanda, cada vez más, servicios de calidad en cualquier momento y en cualquier lugar.

Con el despliegue de LTE, una realidad hoy en día, se alcanzan velocidades de transmisión hasta ahora no disponibles en las redes móviles. Sin embargo, y por las características inherentes de las comunicaciones inalámbricas, los usuarios deben convivir con problemas que merman la calidad percibida, como por ejemplo, altos retardos en el inicio del servicio o frecuentes interrupciones en la reproducción.

El objeto del presente Trabajo Fin de Master es evaluar diferentes arquitecturas y técnicas que permitan desplegar servicios de calidad sobre LTE. En particular, y para soportar servicios de video streaming *multicast*, la arquitectura elegida utiliza el protocolo FLUTE sobre MBMS para el envío de segmentos de video DASH sobre un canal común. Para la recuperación de dichos segmentos, perdidos en la interfaz radio, se utilizan técnicas de corrección de errores FEC y retransmisiones unicast HTTP en los casos en los que dichas técnicas resultan insuficientes. La evaluación se realiza utilizando el código Raptor, técnica FEC estandarizada por 3GPP para dar fiabilidad a las comunicaciones sobre MBMS.

Como resultado de este trabajo, se ha desarrollado una herramienta que permite analizar servicios de video streaming multicast sobre redes LTE, lo que puede ayudar a un operador de red móvil a desplegar este tipo de servicios.

Abstract

Nowadays, video streaming services catch the attention and interest of the society of the information. The growth of wireless communication networks has risen the demand of quality services, every time and everywhere.

With the deployment of LTE, turned into reality by now, it's possible to reach higher transmission rates than the obtained in previous generations of mobile networks. However, by the inherent characteristics of wireless communications, users must coexist with traditional problems that diminish the perceived quality, such as high delays at the service startup or frequent interruptions in playback.

The object of the following work is to evaluate different architectures and techniques to deploy quality services on LTE. In particular, to support *multicast* video streaming services, the chosen architecture uses the FLUTE protocol for sending DASH video segments over MBMS on a common channel. To recover the lost segments on the radio interface, error correction techniques (FEC) and HTTP unicast retransmissions are provided in cases where these techniques are insufficient to recover the original information. In the assessment of this behavior, we have used the Raptor code FEC technique standardized by 3GPP to provide reliable communications over MBMS.

As a result of this work, we have developed a tool to analyze multicast video streaming services over LTE networks, which can help a mobile network operator to deploy these services.

Índice general

Capítulo 1: Introducción	11
Capítulo 2: Estado del arte.....	15
2.1 Modelos de distribución de contenidos	15
2.1.1 Content Delivery Networks (CDN)	16
2.1.2 Peer-to-Peer (P2P)	17
2.1.3 Modelos híbridos CDN-P2P	18
2.2 Tecnologías para el envío de streaming.....	20
2.2.1 Tecnologías para servicios de video streaming sobre protocolos no fiables	21
2.2.2 Tecnologías para servicios de video streaming sobre protocolos fiables	23
2.3 Técnicas de corrección de errores.....	26
2.3.1 Protocolos basados en retransmisiones: ARQ y HARQ	27
2.3.2 Técnicas de corrección de errores en el receptor (FEC)	28
Capítulo 3: Definición de arquitectura para la distribución de video en redes LTE	33
3.1 Arquitectura del sistema LTE.....	33
3.2 Arquitectura para distribución de contenidos multimedia	34
3.2.1 Modelo de distribución de contenidos	34
3.2.2 Tecnologías y protocolos de envío en servicios de video streaming sobre LTE.....	36
Capítulo 4: Simulación de flujos de video unicast y multicast en redes LTE con OPNET Modeler ...	38
4.1 Escenario básico de simulación	38
4.1.1 Aplicaciones.....	39
4.1.2 Perfiles	40
4.1.3 Protocolos.....	40
4.1.4 Parámetros de la red LTE.....	41
4.1.5 Otros	41
4.2 Validación del escenario básico de simulación	41
4.3 Análisis y estudio de la arquitectura definida utilizando escenarios de simulación	43
Capítulo 5: Impacto del uso de técnicas FEC en servicios de video streaming multicast sobre LTE ...	48
5.1 Modelo del sistema estudiado.....	49
5.2 Resultados.....	52
5.2.1 Análisis de cobertura frente a AL-FEC code rate	53
5.2.2 Análisis de tasa de servicio frente a cobertura.....	54

5.2.3	Análisis del ISD en el despliegue del sistema LTE.....	55
5.2.4	Análisis de la duración de los segmentos de video protegidos por AL-FEC ...	56
5.2.5	Compromiso AL-FEC Vs PHY-FEC para maximizar la tasa de datos del servicio.	57
Capítulo 6: Conclusiones y líneas futuras.....		58
6.1	Conclusiones	58
6.2	Líneas Futuras.....	60
Referencias		61

Índice de figuras

Figura 1. Redes de distribución de contenidos	16
Figura 2. Sesión RTSP	22
Figura 3. Arquitectura DASH.....	24
Figura 4. Arquitectura códigos RaptorQ	29
Figura 5. Bloque de datos codificados con RaptorQ	30
Figura 6. Arquitectura LTE	34
Figura 7. Arquitectura para servicios de video streaming multicast	36
Figura 8. Parámetros de configuración de la aplicación Video Conferencing.	39
Figura 9. Escenario básico para la simulación de servicios unicast y multicast sobre LTE.....	42
Figura 10. Tráfico recibido por diferentes usuarios de la red LTE.	43
Figura 11. SNR y paquetes perdidos en los diferentes usuarios	45
Figura 12. Escenario sin congestión	46
Figura 13. Escenario con congestión	46
Figura 14. Escenario para un área MBSFN de 7 celdas	50
Figura 15. Arquitectura para simular el servicio de video streaming.	51
Figura 16. Cobertura Vs AL-FEC code rate para un área MBSFN de 7 celdas.....	53
Figura 17. Cobertura Vs AL-FEC code rate para MCS 15	54
Figura 18. Tasa de servicio de datos Vs Cobertura en área MBSFN de 7 celdas.	55
Figura 19. Tasa de datos Vs Cobertura para MCS 15	55
Figura 20. Tasa de datos Vs Cobertura para distintos ISDs	56
Figura 21. Tasa de servicio Vs Cobertura para diferentes valores del periodo de protección	56
Figura 22. Tasa de datos máxima Vs Cobertura para AL-FEC/PHY-FEC trade-off	57

Siglas

LTE: Long Term Evolution

DASH: Dynamic Adaptive Streaming over HTTP

FLUTE: File Delivery over Unidirectional Transport

MBMS: Multimedia Broadcast Multicast Service

MBSFN: Multicast Broadcast over Single Frequency Networks

CDN: Content Delivery Networks

VoD: Video on Demand

MPD: Media Presentation Description

FDT: File Delivery Table

PHY-FEC: Physic-layer Forward Error Correction

AL-FEC: Application-layer Forward Error Correction

eNB: eNodeB

MCS: Modulation and Coding Scheme

ISD: Inter Site Distance

BLER: Block Error Rate

Capítulo 1

Introducción

La creciente demanda de servicios de video streaming en redes móviles ha planteado nuevos retos y desafíos a la comunidad científica. A pesar de que el despliegue de *Long Term Evolution* (LTE) es una realidad hoy día, se exigen todavía soluciones que permitan explotar eficientemente las capacidades del sistema para ofrecer servicios de calidad.

El usuario, tradicionalmente acostumbrado a típicos problemas, como por ejemplo, altos retardos en el inicio del servicio, frecuentes interrupciones en la reproducción o baja calidad del video, percibe que sobre la red móvil, estos síntomas se presentan muy acentuados. Esto se debe a las inherentes características de la red móvil, que soporta una capacidad limitada sobre un canal no fiable que introduce una mayor tasa de error.

Para solucionar parcialmente este problema, se han propuesto mecanismos de adaptación en el nivel de aplicación. La idea consiste en precodificar el video original en diferentes calidades, y dividir posteriormente, cada uno de los videos obtenidos en múltiples fragmentos, también denominados segmentos. De esta forma, el cliente realiza peticiones HTTP para recibir los segmentos que componen la presentación multimedia, y conmuta entre las diversas calidades para adaptarse al ancho de banda existente en la red. Esta cuestión ha sido estudiada en profundidad y ha generado multitud de propuestas propietarias que han convergido en una solución estándar: *Dynamic Adaptive Streaming over HTTP* (DASH) [1].

DASH utiliza los protocolos HTTP/TCP para enviar el flujo de video, por lo que se requiere un canal bidireccional entre emisor y receptor. En servicios *Video On Demand* (VoD), donde un usuario específico solicita contenido previamente almacenado en los servidores, esta solución puede ser adecuada. Sin embargo, en escenarios donde el contenido es generado en directo (Live Streaming), y entregado a un elevado número de usuarios simultáneamente, la solución no es realizable en la práctica. El principal inconveniente es la necesidad de hacer un uso eficiente de los escasos recursos radio. Con este fin, 3GPP ha estandarizado el uso de *Multimedia Broadcast Multicast Service*

(MBMS) [2], una técnica que se basa utilizar un canal común para entregar el mismo contenido a un elevado número de usuarios.

Sin embargo, el uso de MBMS también plantea desafíos. En primer lugar, no es posible utilizar soluciones basadas en HTTP/TCP, puesto que evitamos establecer una conexión por cada usuario utilizando un canal común. Por otra parte, no es posible implementar el mecanismo de adaptación, los múltiples receptores reciben el mismo contenido, lo que exigiría utilizar varios canales *multicast* para implementar esta técnica. Esta solución, aunque posible, no es adecuada si el operador de la red móvil quiere ofrecer distintos canales de video. En definitiva, en este escenario de recursos limitados se requieren otros mecanismos para garantizar la calidad del servicio.

La arquitectura que está bajo proceso de estandarización por el 3GPP para la provisión de servicios Live Streaming sobre MBMS, utiliza el protocolo *File Delivery over Unidirectional Transport* (FLUTE) [3], sobre UDP, para el envío de los segmentos de video DASH sobre MBMS.

Las características del canal radio, y el uso de UDP como protocolo de transporte, hacen que la entrega fiable del contenido en los receptores sea una tarea ardua. Para dar robustez y fiabilidad al sistema, se deben utilizar técnicas de corrección de errores que permitan a cada usuario, con una condición de canal distinta, recuperarse de las posibles pérdidas. Por esta razón, 3GPP ha estandarizado el uso de los códigos Raptor para la transmisión fiable sobre MBMS [4].

Los códigos Raptor protegen los segmentos de video DASH añadiendo on-the-fly símbolos de redundancia, a los símbolos que componen la información original. Cada usuario, con una condición de canal distinta, será o no capaz de recuperar los segmentos de video, en función de los símbolos recibidos.

Dado que cada usuario presenta una capacidad específica para protegerse frente a errores, existen segmentos de video que no se decodifican correctamente en la transmisión sobre MBMS. Es por esta razón, que la arquitectura propuesta por 3GPP para soportar servicios de video streaming sobre MBMS, contempla la recuperación de los bloques de video a través de conexiones HTTP *unicast*.

La idea es aprovechar la infraestructura web existente y la solución estándar DASH (sobre HTTP). De esta forma, se aplica sobre cada segmento de video la codificación Raptor de forma independiente, y se envían los segmentos, ya protegidos, como ficheros del protocolo FLUTE sobre MBMS.

El dispositivo del cliente debe disponer de un cliente FLUTE donde se reciben todos los ficheros de la sesión, así como un cliente DASH que recibirá los segmentos de forma transparente gracias a una caché HTTP intermedia. Es decir, el cliente DASH realizará peticiones HTTP a la caché, que obtendrá los segmentos de video, o bien desde el directorio en el que se almacenan los ficheros FLUTE (si los segmentos se han decodificado correctamente), o bien desde el servidor (si la decodificación ha finalizado sin éxito).

A la hora de desplegar un servicio de video streaming *multicast* sobre LTE, se deben tener en cuenta todos estos aspectos, y también, otros parámetros que influyen directamente en la calidad de servicio percibida por los usuarios. Algunos de ellos son: el esquema de modulación y codificación (MCS) utilizado, la distancia entre estaciones base (ISD) o la longitud de los segmentos de video. En definitiva, se exige un análisis profundo sobre múltiples escenarios y configuraciones para establecer un valor de los parámetros que maximice un determinado parámetro de utilidad.

El resto del presente Trabajo Fin de Master se organiza de la siguiente forma: El Capítulo 2 presenta una revisión del estado del arte sobre modelos de distribución de contenidos, así como tecnologías y protocolos para el soporte de servicios de video streaming. Tras el estudio del estado del arte se elige, en el Capítulo 3, una arquitectura que considere las características intrínsecas de las redes móviles, del sistema LTE y de las características del servicio de video.

El Capítulo 4 se centra en la simulación de una red LTE con la herramienta de simulación OPNET Modeler [5]. Tras establecer un marco de simulación adecuado, se utiliza un escenario básico para realizar un análisis parcial de la arquitectura propuesta. Finalmente, en el Capítulo 5, se presenta un análisis que evalúa la arquitectura completa y que puede ayudar a un operador a desplegar un servicio de video streaming *multicast* sobre LTE.

Capítulo 2

Estado del arte

2.1 Modelos de distribución de contenidos

Hoy en día, los usuarios demandan, cada vez más, servicios de streaming multimedia. Es por ello que el volumen de tráfico en tránsito ha aumentado notablemente y se espera que continúe en crecimiento, así lo atestiguan diferentes estudios de mercado [6][8].

Sin embargo, distribuir contenido a los usuarios es una tarea ardua para la red convencional, que fue concebida bajo en el principio *best-effort*. El protocolo de red IP no garantiza la entrega fiable y ordenada de paquetes, por lo que no es intrínsecamente eficiente para garantizar retardo y ancho de banda en servicios que generan tráfico muy inelástico y cuyo contenido debe ser reproducido en tiempo real.

Otra de las razones que hacen que no se satisfaga la distribución de contenidos multimedia con el esquema cliente/servidor tradicional, se debe a que los servidores deben responder un elevado número de peticiones. Esto se traduce en la sobrecarga del servidor y el descarte de paquetes, lo que merma de forma apreciable la calidad del servicio. Por otro lado, la limitada capacidad de los servidores tampoco permite que estos almacenen grandes volúmenes de contenidos, por lo que estamos en un escenario de contenido no accesible en el que el cliente debe conectarse a un servidor que puede no estar geográficamente cerca, lo que aumenta, aún más, la latencia y la degradación de la calidad.

Por otra parte, el contenido se presenta al usuario en diferentes formatos y desde diversas fuentes, lo que aumenta la complejidad de la distribución dada la heterogeneidad del escenario.

Los modelos de distribución de contenidos surgen para suplir múltiples deficiencias de las arquitecturas tradicionales, las más importantes son

- Baja calidad de video.
- Alto retardo en el inicio del servicio.
- Frecuentes paradas.
- Necesidad de plugins.
- Protocolos inadecuados.
- DRM no soportado.

2.1.1 Content Delivery Networks (CDN)

El primer modelo bajo estudio es el de las denominadas redes de distribución de contenidos o *Content Delivery Networks* (CDN) (ver Figura 1). Estas redes overlay están formadas por un conjunto de servidores espejo o réplica que distribuyen el contenido, inicialmente almacenado en un servidor origen, a través de los múltiples servidores que la componen.

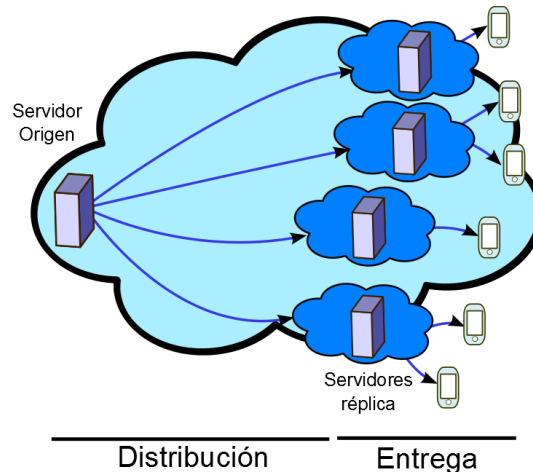


Figura 1. Redes de distribución de contenidos

Este paradigma de distribución presenta importantes ventajas. Algunas de ellas son:

- Reducción de la latencia en términos de RTT por la distribución estratégica de los servidores espejo en la cercanía del usuario.
- Balanceo de carga. Se evita la congestión del servidor origen.
- Solución al problema de capacidad en los servidores. El contenido se replica eficientemente sobre los servidores espejo.

Sin embargo, estas redes también presentan inconvenientes: sus costes de infraestructura y mantenimiento son muy elevados, por lo que se resiente la escalabilidad del sistema (aumenta el precio de los servicios). Por otro lado, las redes CDN convencionales son ineficientes ante el denominado efecto *flash crowd*, problema que surge por el acceso masivo a contenidos populares en intervalos de tiempo acotados (cuello de botella).

Para suplir estas ineficiencias, se han presentado multitud de propuestas con el objeto de mejorar las prestaciones de las redes CDN. Entre las más interesantes se encuentra [9], que presenta *Adaptive Content Distribution Network* (ADCN), una CDN que utiliza nuevos algoritmos de replicación de contenidos para reducir el tiempo de convergencia en eventos *flash crowd*. Otro estudio interesante es el de [10], que presenta *Distributed Content Delivery Network* (DCDN), una solución que propone el uso de una red distribuida de redes CDNs.

Otros trabajos [11][12], más enfocados en la distribución de contenidos en redes móviles, proponen descentralizar el núcleo de la red LTE-A mediante la tecnología *Select IP Traffic Offload* (SIPTO). Esta, permite al usuario conectarse a través del *Home eNodeB* (HeNB), y a través del *Local Gateway* (L-GW), con la red IP. De esta forma, los servidores espejo podrían desplegarse en las cercanías del L-GW, evitándose así, la conexión a través del *Evolved Packet Core* (EPC).

Por último, otro trabajo destacable, nacido en el marco del proyecto MEDIEVAL [13], realiza un estudio *cross-layer* para optimizar la distribución de video. Con este fin, se consideran todas las dependencias entre todas las capas de la pila de protocolos. Dentro de la solución, se ha utilizado una red de distribución de contenidos en un entorno móvil, denominada *Mobile CDN* (mCDN).

2.1.2 Peer-to-Peer (P2P)

A pesar de su evolución, las ineficiencias presentes en las CDN han motivado el estudio de otros modelos de distribución, como por ejemplo, el de las redes *peer-to-peer* (P2P). Estas redes distribuyen el contenido aprovechando la capacidad ociosa de los usuarios, que se conectan entre sí, para entregar o recibir contenido. Al contrario que las redes CDN, las arquitecturas P2P son muy escalables a bajo coste y su rendimiento mejora con el aumento del número de usuarios.

A pesar de solucionar el problema de escalabilidad/coste, presentan otras desventajas que deben ser consideradas. Una de ellas es la posible asimetría en las conexiones. Es decir, los usuarios tienen, habitualmente, un mayor ancho de banda de bajada que de subida, por lo que pueden recibir más datos de los que pueden entregar. Este comportamiento, ya de por sí inequitativo, se acentúa aún más por la capacidad heterogénea de los *peers*.

Otro de los problemas de las redes P2P, es el denominado efecto *churn*, que representa el número medio de nodos que abandonan la red en un determinado intervalo de tiempo. Este, es un factor importante, puesto que la continua entrada y salida de los nodos en el sistema, hace que las tablas de routing sean inconsistentes y

no reflejen la topología real de la red overlay. Este problema, se hace más evidente cuando el abandono de los *peers* no ha sido señalado por problemas de conectividad. En definitiva, la estabilidad del sistema se resiente por el comportamiento dinámico de los nodos, hecho aún más crítico en las redes móviles, donde la variabilidad de este comportamiento aumenta por la movilidad de los usuarios.

Por último, es destacable un problema que afecta a la calidad percibida por el usuario en el uso de servicios de video streaming. En particular, la necesidad de contar inicialmente con un número suficiente de nodos con contenido pre-almacenado (*seeds*), para evitar problemas de retardo en el inicio de la sesión.

En el estudio del modelo P2P, se han revisado los modelos *multi-Tree* y *Mesh* [14]. En el primero de ellos, los *peers* se organizan en múltiples árboles. La entrega del contenido se realiza en *push* desde los nodos de tipo *internal* (con descendientes) a los nodos de tipo *external* (sin descendientes). Sin embargo, este esquema no presenta buen rendimiento ante el efecto *churn*. Por otra parte, en el esquema *mesh*, los *peers* se conectan aleatoriamente formando una red *overlay* (*mesh*). Este sistema es más robusto, sin embargo, experimenta mayores retardos, lo que no es deseable en servicios con requisitos estrictos de retardo. Para mejorar el rendimiento del modelo P2P, se profundiza en [15] sobre estos esquemas, y se propone uno nuevo con un enfoque híbrido *multi-tree/mesh*, denominado *mTreebone*.

2.1.3 Modelos híbridos CDN-P2P

Con la pretensión de obtener los beneficios de ambas tecnologías, surge el concepto híbrido *CDN-P2P*, modelo de distribución que ha generado un encendido interés en la comunidad científica con multitud de propuestas que tratan de optimizar el problema de distribución. Las principales ventajas de este modelo son: elevada escalabilidad del sistema con reducción de costes, estabilidad y fiabilidad en la entrega del contenido gracias al respaldo de los servidores réplica de la CDN, servicio de calidad garantizada (ya sea a través de la CDN, de la red P2P, o de ambas), mejor respuesta al fenómeno *flash crowd*, y reducción del retardo en el inicio de la sesión.

Existen diversas formas de clasificar arquitecturas híbridas CDP-P2P, ya sea por nivel de cooperación, nivel de acoplamiento o tipo de servicio. En términos de nivel de cooperación, se han identificado dos posibles arquitecturas: *peer-aided* CDN (PAC), y *CDN-aided* P2P (CAP). La primera de ellas es la arquitectura PAC, que responde las solicitudes de los usuarios, principalmente, a través de la CDN, mientras que la *overlay* P2P se utiliza para mejorar el rendimiento de la distribución y reducir la congestión de los servidores réplica. Esta estrategia es la utilizada por la mayoría de proveedores de CDN (por ejemplo, ChinaCache y Akamai). La segunda, es la denominada arquitectura

CAP, que delega la distribución a la red P2P y utiliza como respaldo una CDN. La mayoría de proveedores de aplicaciones P2P utilizan este esquema, por ejemplo, PPLive o UUSee.

En el estudio de esta clasificación se destaca el trabajo realizado en [16], donde se muestra una comparación exhaustiva de sistemas PAC como Akamai o LiveSky. En [17], sin embargo, podemos ver un sistema que utiliza una CDN como respaldo (arquitectura CAP). El sistema propuesto ha sido denominado *Proactive, Selective CDN Participation for P2P Streaming* (PROSE).

En términos de nivel de acoplamiento, se han detectado dos posibles modelos: fuertemente acoplados (*tightly-coupled*) y débilmente acoplados (*loosely-coupled*). El primero de ellos es un modelo $1+1$. Es decir, un único sistema CDN puede integrarse con un único sistema P2P. En este sentido, el sistema P2P está estrechamente acoplado a la CDN, que gestiona la construcción de la overlay P2P. El segundo, es un modelo $1+N$, lo que significa que una única CDN puede ofrecer servicios a múltiples sistemas P2P. En este caso, sin embargo, no existe un nivel de integración fuerte, de forma que los sistemas pueden solicitar el uso de la CDN o no. Otra de las ventajas de este esquema es la facilidad que presenta para adaptar el modelo $1+N$ a un modelo $N+N$.

En el estudio de esta clasificación se destaca el sistema *PeerCDN* presentado en [18] como caso de arquitectura *tightly-coupled*. En [19], sin embargo, se presenta el sistema *Web Service-Based content delivery service peering* (WS-CDSP), que responde al modelo *loosely-coupled*.

Para finalizar, y atendiendo al criterio de tipo de servicio, se debe resaltar que en el diseño de las arquitecturas CDN-P2P, se debe tener en cuenta el tipo de tráfico que generan las aplicaciones. Por un lado, los servicios de video streaming con contenido generado en directo (*Live Streaming*) son intolerables frente a retardos, mientras que los servicios de video streaming bajo demanda (*Video on Demand*) no se muestran tan críticos y ofrecen funcionalidades adicionales como interactividad del usuario en el proceso de reproducción.

La documentación bibliográfica utilizada para el estudio de esta cuestión es extensa. Una de las propuestas más interesantes que enfocan el diseño de la arquitectura a servicios *Video on Demand* (VoD), es [20]. En este trabajo, dentro del proyecto europeo COAST, los autores presentan una arquitectura que gestiona información de localización de usuarios, servidores y contenidos para una adaptación flexible entre servicios CDN y P2P.

En cuanto a servicios de tipo *Live Streaming*, la búsqueda bibliográfica también ha reportado estudios de valor. Uno de ellos [21], presenta la arquitectura de *LiveSky*, sistema desarrollado comercialmente por *ChinaCache*.

Tras la presente revisión del estado del arte sobre los modelos de distribución de contenidos, se puede concluir que el uso de arquitecturas CDN, P2P o arquitecturas híbridas CDN-P2P es necesario para poder ofrecer servicios de video streaming de calidad. La reducción de latencia que proporcionan es notable y su uso evita problemas como la congestión de los servidores origen. Sin embargo, esta línea de investigación no está cerrada, el problema de replicación de contenido sobre los servidores es un problema NP-completo, por lo que la búsqueda de la solución óptima debe atender a cada caso de estudio. Por otra parte, y dado que cuentan con mejores prestaciones, el uso de arquitecturas híbridas es recomendable. La integración de ambas tecnologías o la conmutación entre servicios CDN y P2P sigue siendo estudiada hoy en día.

2.2 Tecnologías para el envío de streaming

Los servicios de streaming tienen como objetivo la difusión de contenidos multimedia, de forma que el usuario reproduce el contenido al mismo tiempo que se descarga. Los datos se reciben como un flujo continuo; si el usuario los recibe con una velocidad mayor que la requerida para la reproducción, se guarda el exceso en el buffer, si recibe los datos a una velocidad menor, se producen paradas en el servicio.

Los servicios de streaming objeto de estudio del presente *Trabajo Fin de Master* son:

- *Video On-Demand*: el usuario solicita bajo demanda contenido previamente codificado y almacenado por el proveedor de contenidos.
- *Live Streaming*: el usuario solicita contenido que está siendo generado en tiempo real. Similar a la difusión tradicional de radio y tv.

En la actualidad, existen multitud de mecanismos y protocolos para el envío de streaming (ver Tabla 1). Estas tecnologías se engloban dentro de dos grandes familias: tecnologías que utilizan protocolos de transporte fiable (sobre HTTP/TCP), y tecnologías que utilizan protocolos de transporte no fiables (sobre UDP).

Tabla 1. Tecnologías de envío para video streaming

Sobre HTTP	Sin HTTP
HTTP pseudo-streaming	Real Time Streaming Protocol (RTSP)
HTTP Dynamic Streaming (HDS)	Real Time Multimedia/Messaging Protocol (RTMP)
HTTP Live Streaming (HLS)	Microsoft Media Services (MMS)
Dynamic Adaptive Streaming over HTTP (DASH)	Real Time Transport Protocol (RTP)
Adaptive HTTP streaming (AHS)	Real Time Media Flow Protocol (RTMFP)
Smooth Streaming	File Delivery over Unidirectional Transport (FLUTE)

2.2.1 Tecnologías para servicios de video streaming sobre protocolos no fiables

A) *Real-time Transport Protocol* (RTP)

El protocolo *Real-time Transport Protocol* (RTP) surge ante la elevada demanda de recursos en tiempo real. Está formado conjuntamente con *RTP Control Protocol* (RTCP), un protocolo de control cuya función es proporcionar realimentación para informar sobre la calidad en la distribución de los contenidos.

Este protocolo utiliza UDP como protocolo de transporte. Su función básica es multiplexar varios flujos de datos en tiempo real en un único flujo UDP. Al no incluir control de flujo, se requieren técnicas FEC para corregir los errores en el destino.

B) *Real Time Streaming Protocol* (RTSP)

Tradicionalmente, ha sido uno de los protocolos más extendidos y utilizados en servicios de streaming. *Real Time Streaming Protocol* (RTSP) es un protocolo de control no orientado a conexión, cuyo objetivo es gestionar la entrega de datos en tiempo real (ver Figura 2). Datos que por otra parte, se envían sobre otro protocolo (por ejemplo, RTP).

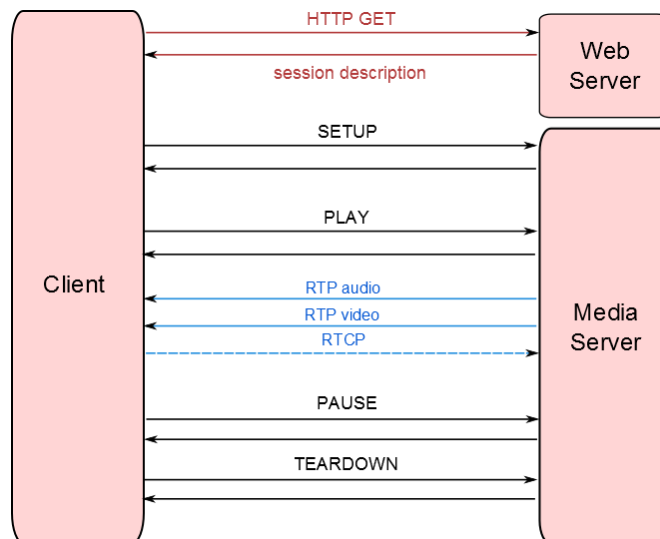


Figura 2. Sesión RTSP

Este protocolo presenta multitud de ventajas. Tiene capacidad multi-servidor, los flujos multimedia que componen la presentación pueden residir en diferentes servidores. Es extensible, se pueden añadir nuevos métodos o parámetros. Es seguro, reutiliza mecanismos de seguridad web, y es independiente del protocolo de transporte, por defecto utiliza TCP para enviar los datos de control del reproductor, pero esto no obligatorio.

A pesar de sus similitudes con HTTP, RTSP presenta diferencias importantes:

- Introduce nuevos métodos y tiene un identificador de protocolo diferente.
- Mantiene una sesión asociada a un identificador. Mantiene el estado de la conexión.
- Tanto servidor como el cliente pueden lanzar peticiones.
- Los datos son transportados por un protocolo diferente.

A pesar de sus ventajas, los sistemas actuales se decantan, cada vez más, por otras tecnologías. Esto se debe, en parte, a la estandarización de DASH, solución que utiliza HTTP/TCP. Pero también, a los problemas que tienen los flujos RTSP para atravesar *firewalls* (los flujos HTTP sin embargo, son difíciles de bloquear). Por otra parte, RTSP es un protocolo que mantiene el estado; establecida la conexión, el servidor envía un flujo de datos continuo hasta que se entrega el video o hasta que el usuario finaliza la sesión. Por esta razón, el usuario puede descargar más contenido del que actualmente necesita, lo que no es deseable.

Por último, y dado que no es una solución definida sobre HTTP, no reutiliza la infraestructura web existente. En particular, las cache HTTP, tan importantes para reducir la carga de los servidores.

C) File Delivery over Unidirectional Transport (FLUTE)

File Delivery over Unidirectional Transport (FLUTE) es un protocolo concebido para entregar ficheros a un número elevado de usuarios. Actualmente, está siendo utilizado para enviar contenido audiovisual, pero también es válido en otros escenarios, por ejemplo, para actualizar el software de múltiples usuarios simultáneamente. A pesar de que puede ser utilizado en aplicaciones unicast o p-t-p (*point-to-point*), su uso es beneficioso en comunicaciones *multicast* o p-t-m (*point-to-multipoint*).

Una sesión FLUTE se identifica por el par (*source IP address*, TSI), donde TSI es el identificador de sesión de transporte o *Transport Session Identifier*, un entero que identifica una determinada sesión de entre todas las activas en el emisor. Cuando una sesión transporta múltiples objetos, cada uno de ellos se identifica por un *Transport Object Identifier (TOI)*. Este identificador, también de tipo entero, adquiere valores mayores que cero. El TOI cero está reservado para el envío de instancias de un fichero de metadatos llamado *File Delivery Table (FDT)*.

Este fichero de metadatos está expresado en *eXtensible Markup Language (XML)*, y describe las propiedades de los objetos transmitidos, así como las características de la sesión. El FDT se transmite como un objeto más de la sesión FLUTE. De esta forma, se envían nuevas actualizaciones o instancias del FDT al receptor con el objeto de que pueda interpretar la información. Como se puede observar, esto es idóneo para servicios *Live Streaming*.

Además, el protocolo FLUTE incluye la opción de proteger el contenido ante posibles pérdidas con técnicas de corrección de errores. Esto se debe a que utiliza el protocolo no fiable UDP, lo que lo hace idóneo en entornos inalámbricos con pérdidas elevadas.

2.2.2 Tecnologías para servicios de video streaming sobre protocolos fiables

A) Dynamic Adaptive Streaming over HTTP (DASH)

Dynamic Adaptive Streaming over HTTP (DASH) es una solución desarrollada por MPEG (*Moving Picture Expert Group*) y publicada como ISO/IEC 23009-1:2012 [1] en abril de 2012. En la actualidad, la comunidad científica está centrando muchos esfuerzos en el estudio de esta tecnología, que se ha erigido como la solución más discutida y plausible para streaming adaptativo.

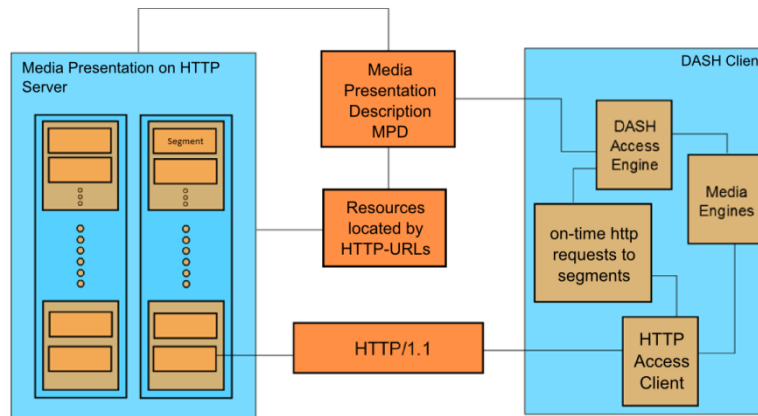


Figura 3. Arquitectura DASH

Esta técnica unicast sobre HTTP, permite que el cliente se adapte al estado de la red conmutando entre diferentes calidades de video. Con este fin, el contenido es previamente codificado a múltiples tasas de transmisión y dividido en pequeños fragmentos, también denominados “segmentos”. Estos segmentos, pueden replicarse sobre una CDN, lo que garantiza mayor accesibilidad y menor latencia en el acceso al contenido.

Los segmentos que comparten la misma tasa y resolución, forman parte de una misma representación de la presentación multimedia. El cliente conmuta entre representaciones de distinta calidad para adaptarse al ancho de banda existente, y evita así, interrupciones en la reproducción del contenido o retardos indeseados.

La capacidad de adaptación del cliente DASH es clave para esta tecnología. Sin embargo, y al contrario de lo que podría parecer, esta cuestión ha generado un encendido debate del que se obtienen multitud de propuestas para encontrar un algoritmo de adaptación óptimo. Para seleccionar una representación de video a una determinada tasa, debe estimarse, en primer lugar, el ancho de banda existente. Sin embargo, los mecanismos de TCP para esta estimación producen resultados incoherentes. Este problema se acentúa en medios inalámbricos, puesto que la pérdida de paquetes en el interfaz radio, es interpretada por TCP como congestión de la red, lo que puede afectar a la adaptación del cliente.

Las propuestas realizadas por otros autores, modelan la adaptación sobre tres parámetros de entrada: tasa de la representación, nivel del buffer y *throughput* promedio (medido en la descarga de segmentos recientes). Incluir el nivel del buffer en el algoritmo de adaptación puede evitar los problemas de alto retardo en el inicio de la sesión, lo que se consigue seleccionando representaciones con video a tasas reducidas. De la misma forma, y si el nivel del buffer supera cierto umbral, a pesar de detectar

mayor congestión en la red, se puede seguir manteniendo la misma calidad hasta que el nivel del buffer sea inferior a dicho umbral.

Otra cuestión importante en esta tecnología, es la localización de los diferentes segmentos que componen la presentación multimedia. Para ello, se genera un fichero de descripción de la presentación o *Media Presentation Description* (MPD). Este fichero de metadatos describe las representaciones y segmentos que componen la presentación multimedia, así como su interrelación. De esta forma, se localiza unívocamente cada segmento de la presentación multimedia a través de un *Uniform Resource Locator* (URL).

Este conjunto de mecanismos para el envío de streaming es muy adecuado en servicios VoD, pero gracias a la posible actualización del MPD, también es posible ofrecer sobre DASH servicios *Live Streaming*.

La cantidad de documentación existente sobre DASH es inabordable. Sin embargo, conviene destacar alguno de los trabajos más interesantes en esta área. En [22], se propone *Peer-Assisted Dynamic Adaptive Streaming over HTTP* (pDASH), que utiliza un modelo híbrido CDN-P2P para aprovechar la capacidad ociosa de los *peers*. Con esta arquitectura, los segmentos son subdivididos en fragmentos o *chunks*. Así, el usuario que solicita un segmento puede conectarse a múltiples *peers* para recuperar los *chunks* que componen el segmento. La implementación de esta metodología es posible gracias a que la idea es compatible al estándar del MPD.

Otros estudios [23][24], proponen el uso de la codificación H.264/SVC en lugar de H.264/AVC. El uso de SVC (*Scalable Video Coding*) permite minimizar la cantidad de información en los servidores, que almacenan múltiples versiones del mismo contenido a diferentes tasas (excesiva replicación de contenido en las CDN). Esto aumenta la tasa de aciertos o *hit-rate* de la memoria cache.

Para entender cómo se ofrece el servicio *Live Streaming* sobre DASH, resulta muy ilustrativo el trabajo presentado en [25], que describe el proceso desde la generación del contenido, hasta su envío, explicando los retardos existentes. Además, muestra un escenario en el que es posible ofrecer publicidad personalizada al usuario.

B) Soluciones propietarias

El nacimiento y estandarización de DASH responde a la existencia de múltiples tecnologías propietarias (ver Tabla 1). No es objeto del presente TFM profundizar en cada una de ellas, pero si es razonable conocer alguna de las soluciones más interesantes:

- *HTTP Dynamic Streaming* (HDS), es la solución propietaria de Adobe, y por tanto, la distribución de contenido multimedia está ligada al software Adobe Flash Access. Como sugiere su nombre, la tasa de transmisión es adaptable, para lo que requiere herramientas que le permitan preparar los contenidos (por ejemplo, fragmentación). Entre las funciones que ofrece, se destaca su capacidad para cambiar la calidad del video en función el ancho de banda, o su capacidad para desplazarse en un video de larga duración sin necesidad de descargarlo primero.
- *HTTP Live Streaming* (HLS) es un protocolo propietario implementado por Apple. Por tanto, su uso es exclusivo en los productos de la compañía que lo soportan. Al igual que las soluciones anteriores, prepara la presentación multimedia fragmentando el contenido en segmentos, lo que le permite ofrecer un streaming adaptativo. Una de las diferencias con respecto a otras soluciones, es la opción de cifrar cada segmento y crear un archivo clave.
- *Smooth Streaming* es una solución de Microsoft que ofrece streaming adaptivo para clientes Silverlight entre otros. Para ello codifica el contenido con la herramienta Expression Encoder en diferentes calidades. Cuando uno de sus servidores recibe una petición sobre un contenido, se crean los fragmentos de video (con el contenedor de formato MP4) en función de las características del equipo del cliente y del ancho de banda. También presenta funcionalidad adicional interesante, como por ejemplo, la posibilidad de agregar cifrado de administración de derechos digitales (DRM).

2.3 Técnicas de corrección de errores

Las técnicas de corrección de errores son necesarias para garantizar que la información llegue correctamente desde el transmisor hasta el receptor o receptores. En función de la capacidad de la red, de la calidad del canal, o de los requisitos de las aplicaciones, será más apropiado elegir una u otra técnica de corrección.

Existen dos principales técnicas de corrección de errores: técnicas basas en retransmisiones como *Automatic Request For Repeat* (ARQ) y *Hibrid Automatic Request for Repeat* (HARQ); y técnicas de corrección de errores en el destino o hacia delante, i.e, técnicas *Forward Error Correction* (FEC).

2.3.1 Protocolos basados en retransmisiones: ARQ y HARQ

A) *Automatic Repeat reQuest (ARQ)*

Automatic Repeat reQuest (ARQ) es un protocolo concebido para detectar errores. Para la detección añade redundancia (CRC) sobre cada bloque de información para comprobar si se ha producido algún error. El receptor no intenta corregirlo, sino que pide la retransmisión del bloque. Para ello, es necesario un canal de *feedback* que le permita comunicar al emisor si el bloque ha llegado correcta o incorrectamente. La tasa de error de este protocolo, es la denominada probabilidad de pérdida P_M , es decir, no detectar el error a pesar de que se ha producido. Los esquemas ARQ proporcionan una tasa de error baja con un decodificador de baja complejidad.

Esta técnica no es adecuada en servicios con requisitos estrictos de retardo y en medios inalámbricos, donde se produce un número elevado de retransmisiones en usuarios con mala recepción de la señal (nivel de señal a ruido insuficiente).

B) *Híbrid Automatic Repeat reQuest (HARQ)*

Los sistemas *Híbrid Automatic Repeat reQuest (HARQ)* integran técnicas ARQ y FEC. En sistemas ARQ puros, la probabilidad de perder un bloque depende únicamente del *Bit Error Rate (BER)* del canal y de la longitud del bloque. La integración de ambas técnicas, puede reducir el número de retransmisiones y aumentar el rendimiento, sin embargo, también aumenta la complejidad del transmisor y receptor.

Existen dos formas de implementar técnicas HARQ. En la primera de ellas, el emisor envía información, tanto para la detección (*Error Detection, ED*), como para la corrección (FEC). Cuando el receptor recibe un bloque, primero analiza el FEC para intentar corregir los posibles errores. Tras lo anterior, usa el ED para detectar si el bloque continúa siendo erróneo. Si la calidad del canal es mala, solo podrá corregir algunos errores y solicitará muchas retransmisiones. Si la calidad del canal es buena, las técnicas FEC serán suficientes para decodificar el bloque con éxito sin retransmisiones.

La segunda implementación no reduce el *throughput* cuando la calidad del canal es buena. Esto se consigue reduciendo el tamaño del código ED y aumentando la del código FEC. Sin embargo, esta técnica reduce el ancho de banda útil para transmitir la información. La metodología que utiliza es la siguiente: primero envía un bloque de datos y utiliza ED para detectar errores, si los hubiera se realiza una retransmisión con parte del FEC. Si al utilizar el código ED sigue detectando el bloque como erróneo, se produce una nueva retransmisión con otra parte del código FEC. Las dos partes se

combinan y se vuelve a utilizar el código ED para detectar errores. Este proceso se itera hasta enviar todo el código FEC.

Por último., existe una tercera implementación HARQ que genera códigos FEC diferentes en cada retransmisión.

2.3.2 Técnicas de corrección de errores en el receptor (FEC)

Existen códigos FEC tanto a nivel físico (*Physic-layer Forward Error Correction*, PHY-FEC), como a nivel de aplicación (*Application-layer Forward Error Correction*, AL-FEC). En este trabajo, se presentarán las técnicas AL-FEC.

Lo más habitual en transmisiones sobre canales ruidosos es utilizar códigos bloque, que como sugiere su nombre, dividen el objeto que solicita el usuario en múltiples fragmentos (*source blocks*). Cada uno de estos bloques se subdivide a su vez, en k símbolos de carga útil (*source symbols*), donde el tamaño de símbolo es constante dentro del bloque. A continuación, se generan r símbolos de redundancia (*repair symbols*) que son añadidos a los k *source symbols*. Así, cada usuario, con una condición de canal distinta, decodificará el video en función de la cantidad de redundancia introducida en el transmisor, y de los n símbolos recibidos (*encoding symbols*).

En la actualidad, la solución estandarizada por el 3GPP para comunicaciones que aplican técnicas de corrección de errores sobre *Multicast Broadcast Multicast Service* (MBMS), se basa en el uso de los códigos Raptor. A pesar de ello, se debe destacar que hoy en día existen otras soluciones que mejoran el rendimiento de estos códigos. Estas novedosas técnicas FEC son los códigos RaptorQ [26] y RS+LDPC-Staircase [27], ambas soluciones han sido propuestas como estándar.

A) Códigos Raptor

Los códigos Raptor [28] son códigos fuente, lo que significa que el codificador puede generar on-the-fly tantos símbolos codificados (*encoding symbols*) como desee, a partir de los *source symbols* del bloque de datos. La solución estandarizada contempla el uso del código sistemático, por lo que incluyen la entrada no modificada en la salida codificada.

Estos códigos dividen el contenido en bloques de k *source symbols*. Posteriormente, generan n *encoding symbols* ($n > k$), y añaden los r *repair symbols* ($n - k$) al payload del paquete. De esta forma, el decodificador recupera el bloque FEC (*source block*) si recibe un conjunto de *encoding symbols* levemente superior a k .

El rendimiento del código Raptor se describe como la probabilidad de fallo del decodificador, que se modela en función del número de *source symbols* que componen

el bloque FEC y los *encoding symbols* recibidos. Así, la probabilidad de fallo del decodificador se calcula como:

$$P(f_{RC} | n) = \begin{cases} 1 & \text{if } n < k \\ 0.85 * 0.567^{n-k} & \text{if } n \geq k \end{cases} \quad (1)$$

Como se deduce de la ecuación anterior, la robustez del sistema depende de la redundancia añadida en la transmisión. Por otra parte, añadir excesiva redundancia limita el ancho de banda útil para la transmisión del video. Esto es especialmente relevante en redes móviles donde el ancho de banda es limitado. Es decir, se debe minimizar la redundancia y maximizar el ancho de banda útil en redes con recursos limitados.

B) Códigos RaptorQ

Recientemente, se ha propuesto una evolución de los códigos Raptor. Esta solución, denominada RaptorQ, presenta muchas similitudes con respecto a predecesora (código fuente, sistemático, etc.), pero también algunas diferencias importantes.

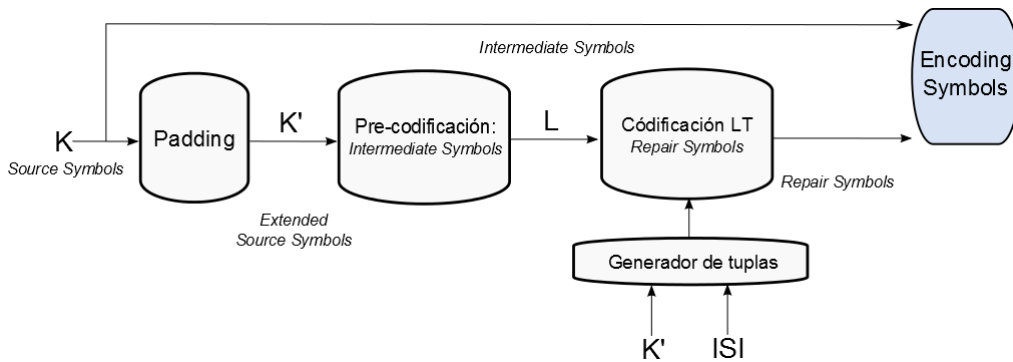


Figura 4. Arquitectura códigos RaptorQ

En la Figura 4 se muestra su arquitectura. A los k *source symbols* que componen cada *source block*, se añaden símbolos de padding. El bloque de salida está compuesto por k' símbolos y se denomina *extended source block*. Estos símbolos de padding se utilizan para aumentar la velocidad de los procesos de codificación y decodificación. Además, no se requiere su envío, puesto que transmisor y receptor saben que cantidad deben añadir dado k , son datos tabulados.

El siguiente paso en la codificación es generar L símbolos intermedios o *intermediate symbols* (donde $L > K'$). Esta pre-codificación, es necesaria para garantizar que el código es sistemático. Una vez que se han generado estos símbolos intermedios, se aplica un código LT, que genera los r *repair symbols*. Como LT es un código fuente, se generan tantos *encoding symbols* como se desee, por lo que RaptorQ también recibe esta propiedad.

La generación de los *repair symbols* se produce al realizar una operación XOR sobre un subconjunto de *intermediate symbols* determinados. La elección de este subconjunto se obtiene del generador de tuplas, cuyos parámetros de entrada son el número k' y un identificador de símbolo llamado *Internal Symbol ID* (ISI).

La Figura 5 muestra un bloque de datos codificado con RaptorQ. Se muestra que está compuesto por un número k de *source symbols*, identificados por un ISI y por un *Encoding Symbol Identifier* (ESI), ambos con el mismo subíndice. A continuación se añade el padding, pero estos símbolos no forman parte de los *encoding symbols* de salida, sino que solo se utilizan internamente para mejorar el rendimiento de los códigos. Es por ello, que no se identifican con un ESI, y la razón por la que existen dos tipos de identificadores.

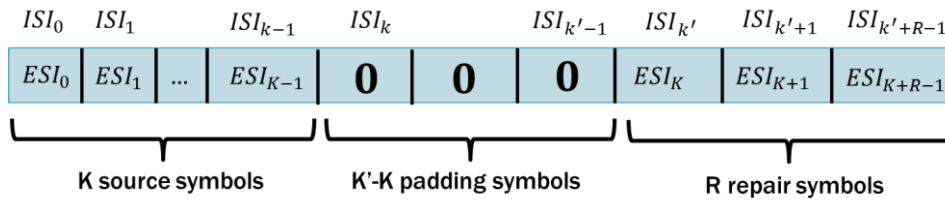


Figura 5. Bloque de datos codificados con RaptorQ

El rendimiento de estos códigos es superior al de su predecesor y han sido propuestos como estándar. La probabilidad de fallo del decodificador en RaptorQ es:

$$P(f_{RC} | n) = \begin{cases} 1 & \text{if } n < k \\ 0.01 * 0.01^{n-k} & \text{if } n \geq k \end{cases} \quad (2)$$

C) Códigos RS+LDPC-Staircase

OPEN-FEC es un proyecto cuya iniciativa es ofrecer una solución de código abierto (escrito en C) que sea una alternativa real a los códigos RaptorQ. Esta solución se basa en una combinación de códigos *Reed-Solomon* (RS) y *LDPC-Staircase*. A diferencia de Raptor, y a semejanza de RaptorQ, no son códigos binarios, esto se debe al uso de RS, que realiza operaciones sobre campos o cuerpos finitos de Galois $GF(2^8)$.

Los códigos *Low Density Parity Check* (LDPC) son sistemáticos y se basan en una matriz que define las relaciones entre los símbolos fuente y de paridad. En esta matriz, k es el número de símbolos fuente y $n-k$ es el número de símbolos de paridad. Por tanto, el tamaño de la matriz es $n \times (n - k)$. De esta forma, la matriz de paridad esta subdivida en dos matrices; mientras que la parte izquierda se refiere a los símbolos fuente, la parte derecha hace referencia a los símbolos de paridad. En función de la

estructura de esta matriz se distinguen dos tipos de código LDPC: regulares e irregulares.

En los regulares todas las filas y columnas de la matriz tienen el mismo grado, donde grado se refiere al número de elementos no nulos de una fila o columna. Los códigos *Staircase* son irregulares y reciben su nombre dado que su submatriz de paridad tiene forma de escalera.

$$H_{LDPC} = \left(\begin{array}{cccc|cccc} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \quad H_{LDPC_Staircase} = \left(\begin{array}{cccc|cccc} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right)$$

El otro código en el que se basa RS+LDPC-Staircase es Reed-Solomon, un código bloque que crea símbolos a través de secuencias de m bits. Cada palabra código, está formada por n símbolos: k símbolos fuente y r símbolos de redundancia.

Como se comentó anteriormente se basan en el uso campos finitos de Galois. Un campo finito tiene la propiedad de que las operaciones sobre elementos del campo dan como resultado un elemento del propio campo. Estos tienen un número de elementos $q = p^n$, donde p es un número primo y n un entero positivo. Los elementos de $GF(p^m)$ se representan como polinomios de grado menor que m . Es decir:

$$GF(p^m) = \{\lambda_0 + \lambda_1x + \lambda_2x^2 + \dots + \lambda_{m-1}x^{m-1}\}$$

Donde los coeficientes λ_m existen en Z_p , y donde Z_p es el cuerpo de elementos formado por los enteros módulo p .

Dado que para RS se trabaja en campos $GF(2^m)$, donde $m \in [2, 16]$. El número máximo de símbolos n que puede considerarse es igual al tamaño del campo, es decir, $q - 1 = 2^m - 1$. Para $m=8$ nos encontramos con una aplicación interesante, puesto que cualquiera de los elementos de un vector en estos códigos es, en sí mismo, un vector de 8 bits (un byte).

Sin entrar en más profundidad sobre este concepto matemático, puesto que no es objeto de este TFM, si conviene resaltar que los códigos RS pueden definirse de dos formas distintas: en la primera de ellas, los k símbolos del mensaje se ven como coeficientes de un polinomio $p(x)$ de máximo grado $k-1$, donde estos coeficientes pertenecen a un cuerpo finito de orden q . Evaluando el polinomio en $n > k$ puntos se obtienen los n elementos que forman la palabra código. En la segunda de ellas, los símbolos de codificación son los coeficientes del polinomio de salida $c(x) = p(x) \times$

$g(x)$. Donde $p(x)$ es el polinomio mensaje de máximo grado $k-1$ y $g(x)$ un polinomio generador de máximo grado $n-k$.

Capítulo 3

Definición de arquitectura para la distribución de video en LTE

3.1 Arquitectura del sistema LTE

Para ofrecer soluciones orientadas a la optimización de los servicios multimedia y multiusuario en redes LTE es necesario conocer la arquitectura del sistema LTE. En la Figura 6 se muestra dicha arquitectura. Sin embargo, en este documento, profundizaremos sobre las nuevas capacidades y entidades que se han añadido a la arquitectura EPS para ofrecer servicios multicast. El resto de entidades, propias de la arquitectura general de LTE se dan por conocidas.

MBMS (Multicast Broadcast Multimedia System) es un servicio p-t-m propuesto por el 3GPP [2], que permite transmitir datos desde un emisor y hasta múltiples receptores empleando un canal común. Gracias a MBMS es posible hacer un uso eficiente de los recursos radio en un marco flexible. Es decir, la asignación de recursos permite soportar servicios MBMS y servicios unicast simultáneamente, donde los servicios MBMS pueden utilizar hasta un máximo de 6 subtramas por trama radio.

En la actualidad, existen técnicas para mejorar eficiencia espectral de MBMS. Una de estas técnicas es la denominada MBSFN (Multicast / Broadcast over Single Frequency Networks). Utilizando esta opción, presente en LTE, todas las estaciones base del área MBSFN cooperan para transmitir la señal sincronizándose con precisión en el tiempo. Además, se mejora el *Signal to Interference plus Noise Ratio* (SINR) del usuario, ya que MBSFN consigue que la interferencia destructiva de una transmisión convencional, se convierta en interferencia útil o constructiva.

Las nuevas entidades de red que permiten ofrecer este servicio son:

- BM-SC (Broadcast/Multicast Service Center): Es el punto de acceso a los proveedores de contenidos. Realiza funciones de control de acceso para la conexión de los terminales al servicio MBMS e incorpora mecanismos de confidencialidad y protección de la integridad de los mensajes transmitidos sobre MBMS.

- MBMS-GW (Multimedia Broadcast Multicast Services Gateway): se encarga de distribuir los paquetes de la sesión MBMS a todos los eNBs bajo el mismo área de servicio. Gestiona la sesión MBMS y realiza funciones relacionadas con la tarificación del servicio MBMS.

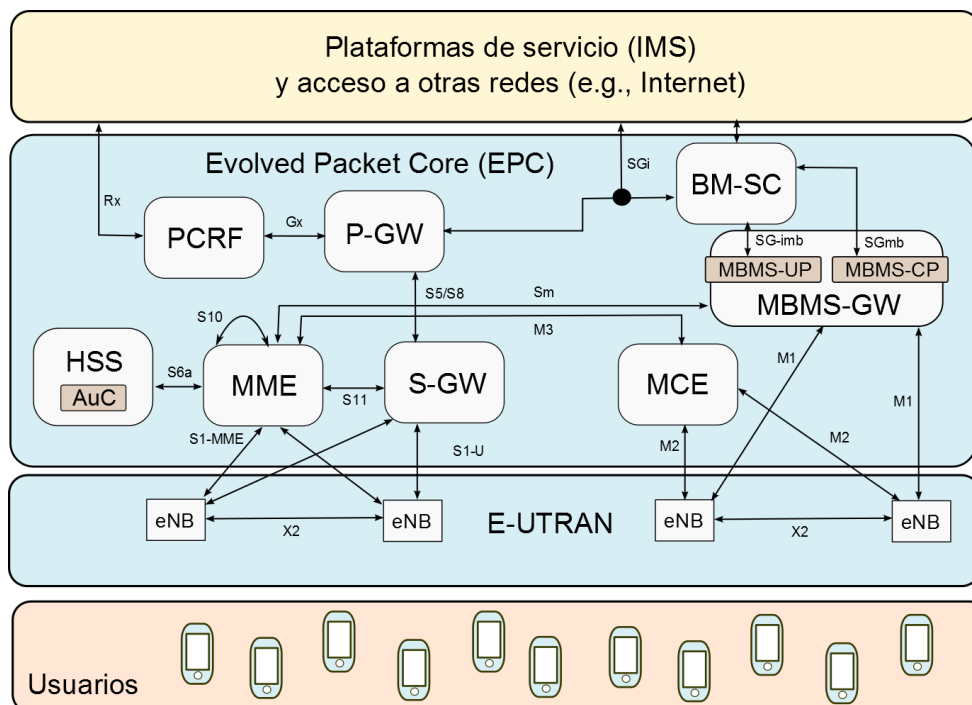


Figura 6. Arquitectura LTE

- MCE (Multi-cell/Multicast Coordination Entity): es la entidad lógica encargada de la asignación de los recursos radio utilizados por los eNBs en cada área MBSFN. Para ello establece, entre otros parámetros, el esquema de modulación y codificación (MCS) utilizado en la transmisión multicast. Esta entidad puede formar parte del eNB o puede ser un bloque autónomo dependiendo de la implementación escogida.

3.2 Arquitectura para distribución de contenidos multimedia

3.2.1 Modelo de distribución de contenidos

Tras la revisión del estado del arte, se concluye que el modelo híbrido CDN-P2P integra las fortalezas presentes en ambas tecnologías y da solución a las debilidades de los modelos puros.

A) CDN-P2P en redes móviles

En un modelo de distribución P2P, los usuarios se conectan entre sí intercambiándose el contenido multimedia. Este contenido ha sido previamente

dividido en segmentos o *chunks*. En redes móviles, donde existe un ancho de banda limitado y la tasa de rotación de los *peers* es muy elevada (debido a su movilidad), este esquema no ofrece un buen rendimiento.

Vistas las desventajas de las CDN en el Capítulo 2, la solución es un esquema híbrido CDN-P2P en el que los usuarios contribuyen muy poco en el enlace de subida. Este esquema implementa el modelo P2P a nivel de servidor réplica.

De esta forma, la arquitectura propuesta en [29] está formada por dos elementos principales: *CDN replica peers* y *mobile peers*. Los servidores de la CDN pueden intercambiarse directamente el contenido, mientras que por otra parte, los usuarios pueden recibir datos multimedia desde múltiples servidores simultáneamente. Cuando un *mobile peer* solicita un contenido, contacta inicialmente con el servidor réplica más cercano y le solicita información de la localización de dicho contenido en los servidores de la CDN. Tras recibir esta información, el usuario planifica cuando y de qué *CDN replica peer* va a recibir el contenido.

B) Telco-CDN

Hoy en día, importantes proveedores de servicios de Internet (ISPs) han desplegado su propia CDN dentro de su infraestructura de red. Estos proveedores, no acostumbrados a operar con repositorios y data-centers, se han visto obligados a profundizar en estas tecnologías para aprovechar una clara oportunidad de negocio.

En una Telco-CDN [30], la gestión de la CDN recae en el operador de red, que posee la infraestructura de las redes overlay y de las redes subyacentes. Así, la información adicional que se obtiene de la red subyacente puede ser utilizada para encontrar una mejor solución al problema de distribución.

A pesar de que tradicionalmente se ha implementado una estrategia de reemplazo de contenidos en la memoria caché de los servidores, recientemente se ha vuelto a estudiar la posibilidad de que el operador decida como replicar los contenidos en los servidores. Principalmente, gracias a una mejor distribución del contenido por un mayor control del operador sobre la misma (uso de la información de la red subyacente).

Otra razón son los interesantes avances en algoritmos que predicen la popularidad de los contenidos. En esta discusión, han surgido estudios que modelan el problema de distribución como un problema de localización *k-PCFLP*. Para resolver este problema se utiliza un algoritmo genético (GA), que implementado en un entorno MapReduce,

encuentra el emplazamiento óptimo gracias al estado de la red o al índice de popularidad del contenido.

El auge de las Telco-CDN no es necesariamente peligroso para los proveedores de CDN tradicionales. De hecho, existen mecanismos de interconexión entre ambas a través puntos de presencia o *Point-of-Presence* (PoP). Cuando el operador no dispone de un contenido en sus repositorios locales, accede a través del PoP a la red global CDN.

3.2.2 Tecnologías y protocolos de envío en servicios de video streaming sobre LTE

La solución que está actualmente bajo proceso de estandarización por el 3GPP utiliza DASH en servicios Live Streaming sobre MBMS. Esta solución utiliza el protocolo FLUTE para enviar los segmentos de video DASH, como objetos de FLUTE sobre UDP.

La Figura 7 muestra la arquitectura propuesta. En ella se identifica un cliente FLUTE/DASH que recibe, inicialmente, un fichero de metadatos MPD que contiene la dirección URL de los segmentos disponibles. Estos segmentos, son enviados en una sesión FLUTE sobre UDP a los múltiples receptores que participan en ella. Junto a los objetos que llevan los segmentos DASH, se envían instancias del FDT para describir la sesión FLUTE a los usuarios.

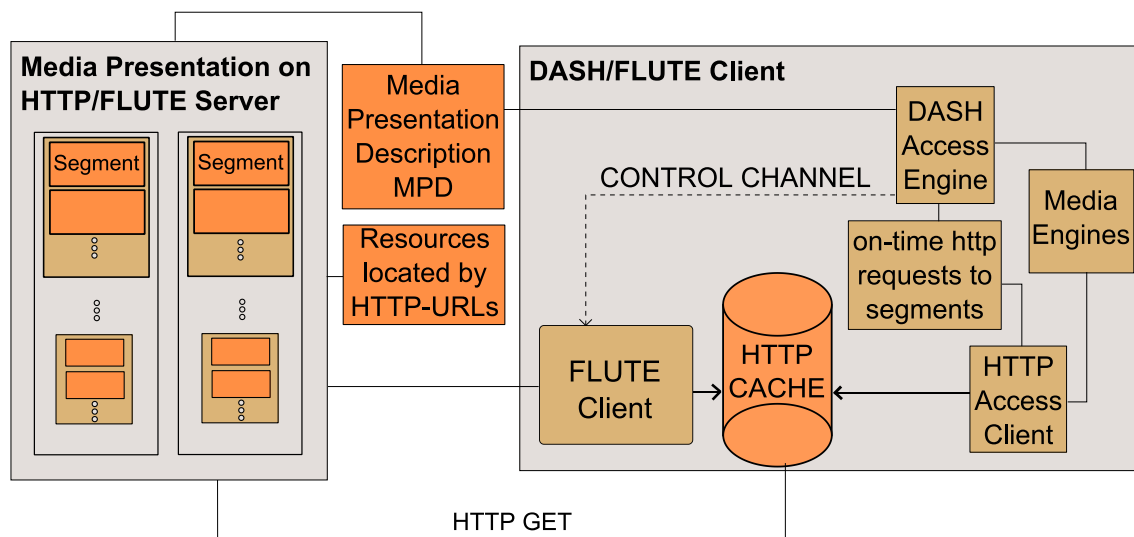


Figura 7. Arquitectura para servicios de video streaming multicast

La transmisión de los objetos FLUTE se realiza sobre MBMS hasta los múltiples destinatarios que participan en la sesión. Los receptores mantienen una base de datos con las instancias FDT recibidas. Recuérdese, que los receptores se distribuyen uniformemente sobre el área de cobertura de la red móvil, por lo que la tasa de pérdidas experimentada en cada usuario es distinta. Por esta razón, los usuarios no

mantienen la misma información de la sesión FLUTE, no todos reciben la instancia FDT en un determinado instante de tiempo.

Una vez que se reciben los ficheros en el cliente FLUTE del terminal móvil, estos pasan por una caché HTTP intermedia, y posteriormente, el cliente DASH recupera los segmentos de video de esta caché. Es decir, el cliente recibe los segmentos de video DASH sin ser consciente de que están llegando en una sesión FLUTE (de forma transparente), y es la caché, la encargada de solicitar el contenido que no se ha recibido correctamente en la sesión FLUTE multicast, a través de solicitudes HTTP unicast con el servidor origen.

Esta arquitectura es flexible y robusta, puesto que permite recuperar segmentos de video descartados en la sesión FLUTE sobre UDP, a través de una conexión unicast HTTP. Es importante destacar en este punto, que FLUTE puede utilizar las técnicas FEC presentadas en el Capítulo 2, por lo que la probabilidad de perder un segmento, dependerá, entre otros parámetros, de la cantidad de redundancia añadida en cada bloque de información útil.

Capítulo 4

Simulación de flujos de video unicast y multicast en redes LTE con OPNET Modeler

Para simular la transmisión de flujos de video unicast y multicast en redes LTE, se ha utilizado la herramienta de simulación OPNET Modeler. La elección de esta herramienta se justifica por varias razones, una de ellas es la facilidad que muestra para analizar diferentes niveles de la torre de protocolos, lo que permite la cooperación y coordinación de varios grupos de trabajo especializados. Así, tanto el grupo de la Teoría de la señal y las Comunicaciones (TSC) de la Universidad Carlos III, como el grupo de redes de la Universidad Politécnica de Madrid han trabajado de forma conjunta en la configuración de diversos escenarios de simulación con el objetivo de obtener resultados coherentes y de valor para cada grupo.

En este capítulo, se detallará la configuración de un escenario básico de simulación de una red LTE.

4.1 Escenario básico de simulación

El escenario que se ha definido simula una red compuesta por tres estaciones base (eNB), el núcleo de la red (EPC) y 47 usuarios caracterizados en ocho grupos. Sobre este escenario se simulan flujos de video streaming para comunicaciones unicast y multicast.

La configuración del escenario requiere fijar el valor de un elevado número de parámetros de nivel físico, red y transporte. Además, se deben definir múltiples parámetros del proyecto de simulación o la arquitectura de red LTE. En definitiva, la configuración involucra los siguientes elementos:

- Aplicaciones
- Perfiles
- Protocolos
- Parámetros de la LTE
- EPC
- Estaciones base (eNBs)
- UEs y Servidores

4.1.1 Aplicaciones

Las aplicaciones que se han utilizado para el análisis de flujos de video en comunicaciones unicast y multicast son, respectivamente: HTTP y *Video Conferencing*. Por un lado, dentro de las extensas posibilidades que ofrece OPNET para configurar la aplicación HTTP, se ha utilizado la opción Live Streaming. La configuración de este servicio requiere definir que la consulta HTTP se realiza sobre contenido de video, y que este, se descarga y reproduce inmediatamente tras establecer la conexión. Por otro lado, se ha utilizado la aplicación Video Conferencing para streaming *multicast* sobre UDP. En este caso, la aplicación de videoconferencia genera tráfico en sentido cliente-servidor y servidor-cliente, por lo que ha sido necesario estrangular el ancho de banda en el lado cliente-servidor.

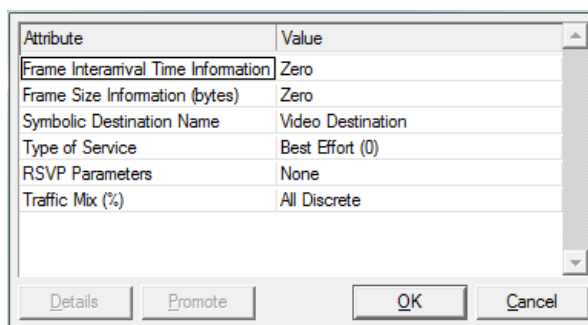


Figura 8. Parámetros de configuración de la aplicación Video Conferencing.

Como ejemplo de ello se muestra en la Figura 8 la ventana de configuración del servicio *Video Conferencing*. En ella se observan diversos parámetros, los más importantes son:

- *Frame Interarrival Time Information* (frames/sec): número de frames por unidad de tiempo.
- *Frame Size Information* (bytes): tamaño de cada frame. En este parámetro se define tanto el tamaño del frame entrante como del frame saliente. Para el escenario de simulación propuesto, se estrangula el tráfico entrante en los servidores:
 - Incoming Stream Frame size (bytes): tamaño del frame entrante. Fijado a cero bytes.
 - Outgoing Stream Frame size (bytes): tamaño del frame saliente.

Por último, se debe resaltar que el tráfico generado no tiene en cuenta ningún criterio de codificación y compresión de video. OPNET se abstrae de esta cuestión, y simplemente genera tráfico de video a una determinada tasa. En HTTP, aplicación elegida para simular servicios de streaming *unicast*, ocurre exactamente lo mismo.

4.1.2 Perfiles

OPNET dispone de los elementos necesarios para permitir una caracterización de usuarios muy flexible. La herramienta permite definir con precisión, los perfiles de tráfico que ejecutan los usuarios. De esta forma, se definen las aplicaciones que utilizan, cuando las utilizan y durante cuánto tiempo. La libertad de la herramienta permite, por ejemplo, definir perfiles que pueden ser ejecutados concurrentemente, y en los que se ejecutan aplicaciones de forma secuencial o en paralelo.

Para el escenario básico de simulación, se han definido ocho perfiles distintos: cuatro para usuarios que ejecutan servicios *unicast* y cuatro para usuarios que ejecutan servicios *multicast*. Cada perfil genera tráfico de video a una determinada tasa, como se puede ver en la Tabla 2

Tabla 2. Perfiles de usuarios en el escenario básico de simulación

Perfiles	Usuarios	Tasa
Multicast 224.0.6.1	8 Users	200 kbps
Multicast 224.0.7.1	7 Users	300 kbps
Multicast 224.0.8.1	11 Users	400 kbps
Multicast 224.0.9.1	11 Users	500 kbps
Unicast 6	2 Users	500 kbps
Unicast 7	3 Users	1,6 Mbps
Unicast 8	2 Users	2,1 Mbps
Unicast 9	3 Users	2,7 Mbps

4.1.3 Protocolos

Los protocolos de nivel de red y transporte que se han configurado son: RIP, PIM-SM y IGMP. Se ha configurado RIP como protocolo de encaminamiento, puesto que el tamaño de la red simulada no requiere que los paquetes den muchos saltos, y por otra parte, se pretendía definir un protocolo diferente al propuesto por defecto en la herramienta de simulación (OSPF). Por otro lado, se ha configurado PIM-SM como protocolo de routing *multicast*. Este protocolo utiliza la información de los protocolos de encaminamiento *unicast* para realizar IP-Multicast. Conviene destacar que no es obligatorio su uso en sesiones MBMS, pero es eficiente para encaminar los paquetes.

Para finalizar, se ha configurado IGMP para la gestión de los grupos *multicast*. La información de este protocolo también es necesaria en el protocolo PIM-SM. La implementación de IGMP en OPNET permite, entre otras cosas, establecer el momento en el que un usuario se une a una sesión multicast, y el momento exacto en el que la abandona. En el escenario básico de simulación, un usuario se unirá exclusivamente al grupo multicast cuyo perfil ejecute.

Otra consideración importante, es que la librería de LTE incluida en la herramienta de simulación, solo permite ejecutar un servicio *multicast* en cada subtrama. Es decir, no se pueden asignar varias subtramas al mismo servicio multicast, ni tampoco se pueden asignar varios servicios multicast a una única subtrama. Esto es una limitación de OPNET, esta restricción no está presente en el estándar del 3GPP.

4.1.4 Parámetros de la red LTE

Para poder obtener resultados coherentes, es necesario definir parámetros físicos de la red LTE. Algunos de estos parámetros, críticos para el buen funcionamiento del sistema, son: el esquema de modulación y codificación (MCS), parámetros para la configuración de las áreas MBMS y MBSFN o definición de las portadoras EPS y MBMS.

4.1.5 Otros

La configuración de todos estos puntos no aplica si: no se activan los protocolos en las interfaces de red correspondientes, no se define en los eNBs la pertenencia al área MBMS, o no se establecen los perfiles en usuarios y las aplicaciones en los servidores. Es decir, además de los puntos que se han resaltado, existen otras muchas consideraciones que exigirían una comunicación más profunda y extensa.

Por otra parte, se debe resaltar que los usuarios de la red móvil se distribuyen aleatoriamente sobre el área de cobertura de la estación base. Sin embargo, para este estudio, no se han ejecutado perfiles de movilidad. Es decir, los usuarios permanecen estáticos durante el tiempo simulado.

4.2 Validación del escenario básico de simulación

A continuación, se valida la configuración anterior con la simulación del escenario básico. La Figura 9 muestra algunos de los aspectos que se han ido citando; el escenario está compuesto por 47 usuarios segmentados en ocho perfiles distintos y se han desplegado tres estaciones base y un núcleo EPC. Junto a la figura, se adjunta una leyenda con el objeto de facilitar la identificación de los perfiles creados.

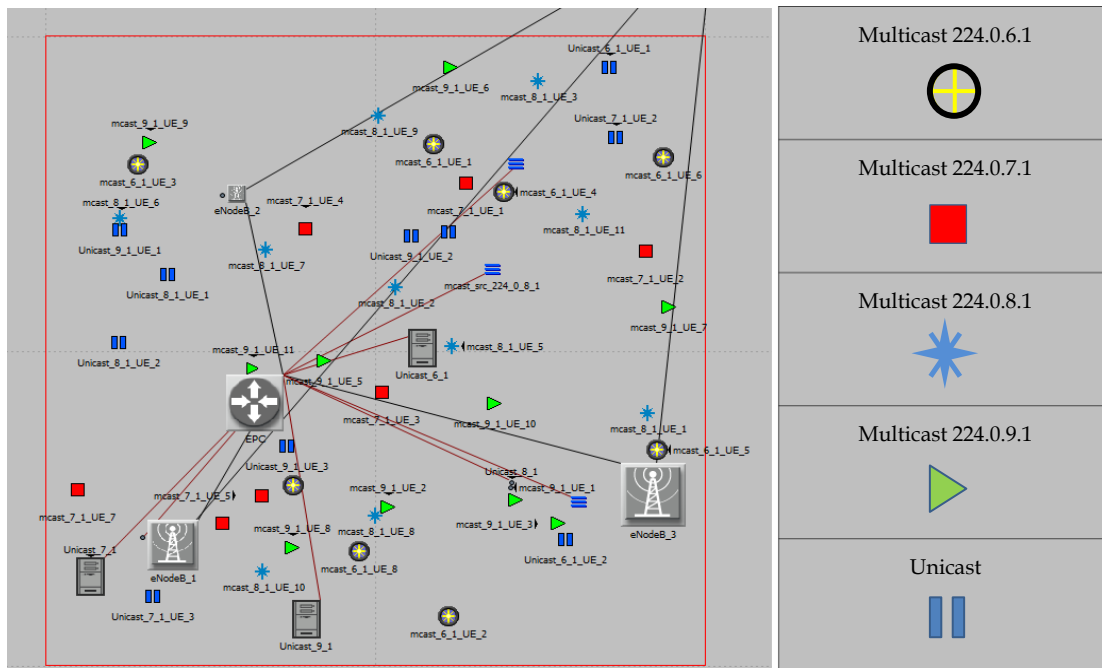


Figura 9. Escenario básico para la simulación de servicios unicast y multicast sobre LTE

A continuación, se va a comprobar que la tasa de video definida a nivel de aplicación es recibida correctamente en los usuarios de la red. Para ello, es importante notar que para el estudio de este caso, los usuarios están próximos a las estaciones base. Es decir, no se esperan pérdidas de paquetes destacables, lo que se debe a que la calidad del canal es buena.

En la Figura 10 se muestra el tráfico recibido en distintos usuarios de la red. Como se puede observar, el tráfico recibido se corresponde con las tasas definidas a nivel de aplicación (ver Tabla 2). Por otro lado, se observa una duración diferente de los servicios unicast y multicast, lo que se debe a que se ha mantenido la sesión multicast durante todo el tiempo simulado, mientras que los usuarios unicast, visualizando otros contenidos, han finalizado su reproducción en un instante de tiempo anterior.

Estos resultados certifican que la configuración de: aplicaciones, perfiles, protocolos, parámetros de la LTE, EPC, estaciones base, UEs y Servidores se ha realizado correctamente. No es objeto del presente Trabajo Fin de Master especificar todos los parámetros que se han utilizado para lograr estos resultados. Para más información se recomienda la lectura de la documentación que incluye la propia herramienta de simulación.

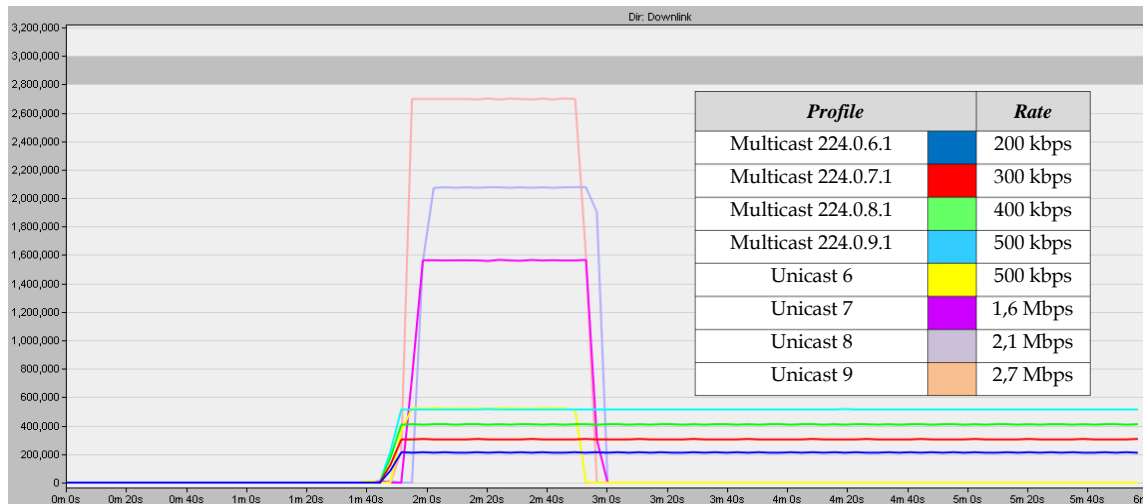


Figura 10. Tráfico recibido por diferentes usuarios de la red LTE.

4.3 Análisis y estudio de la arquitectura definida utilizando escenarios de simulación

En esta sección, se estudiará y analizará parte de la arquitectura definida en el Capítulo 3. Como se ha visto en este documento, es posible simular flujos de video unicast y multicast. Sin embargo, no es posible simular algunos de los mecanismos utilizados en la arquitectura propuesta por el 3GPP. La herramienta no implementa, entre otras cosas, streaming adaptativo DASH o la posibilidad de recuperar video a través de una HTTP caché intermedia.

Es por ello, que un estudio más profundo de DASH y FLUTE se está realizando por otros cauces que no procede explicar aquí. A pesar de estas dificultades, la herramienta ha sido útil para evaluar escenarios con pérdidas de paquetes, y por tanto, para estudiar (en el Capítulo 4) el impacto del uso de técnicas de corrección de errores sobre MBMS.

Tabla 3: Elementos del escenario de simulación

Simulation Scenario		
Profile	Users	Rate
Multicast 224.0.6.1	8 Users	256 kbps
Multicast 224.0.7.1	7 Users	512 kbps
Multicast 224.0.8.1	11 Users	750 kbps
Multicast 224.0.9.1	11 Users	1,5 Mbps
Unicast 6	2 Users	256 kbps
Unicast 7	3 Users	512 Mbps
Unicast 8	2 Users	750 Mbps
Unicast 9	3 Users	1,5 Mbps

En este capítulo, se analizarán los factores que influyen para que un usuario, que está recibiendo contenido de video a través de MBMS, conmute a un servicio unicast para recuperar los segmentos que se han perdido por el interfaz radio en la sesión FLUTE.

Para realizar este análisis se ha utilizado el escenario que se recoge en la Tabla 3. En este caso, los perfiles *unicast* generan el mismo tráfico que los *multicast*. De esta forma, se pueden comparar directamente las pérdidas de paquetes en ambos servicios.

Ahondando en la arquitectura elegida, recuerde que como se ha explicado anteriormente, la transferencia fiable en entornos inalámbricos se consigue utilizando técnicas de corrección de errores en el destino (FEC). No existe la retransmisión de paquetes a través del protocolo FLUTE (UDP), por lo que se protege la información añadiendo redundancia. El problema que conlleva introducir redundancia, es la reducción de la carga útil de cada bloque de información transmitido. Reducir dicha carga, limita el ancho de banda disponible para la transmisión del video codificado.

El video tiene, por tanto, un ancho de banda útil:

$$BW_u = BW_{total} - R$$

Donde R es la redundancia que se introduce en cada bloque de información, para que en caso de error, el decodificador FEC pueda recuperar el bloque original. Existe por tanto, un compromiso entre la capacidad de corrección de errores y el ancho de banda que se destina a la transmisión del video, y por tanto, a la calidad del mismo.

Por otra parte, existe la posibilidad de que no se pueda recuperar la información utilizando únicamente la redundancia ofrecida por la cabecera FEC. En ese caso, se deberá utilizar el servicio unicast para recuperar el bloque de información perdido. El número de paquetes retransmitidos, dependerá del tamaño de la redundancia FEC y de las condiciones de canal que tenga cada usuario en concreto (es decir, del nivel señal a ruido SNR). En la Figura 11 se muestra este comportamiento. A medida que se aleja el usuario de la estación base, y por tanto, del área de cobertura, peor es la recepción de la señal en el usuario, y por tanto, mayor la pérdida de paquetes.

A pesar de que los usuarios permanecen estáticos durante el tiempo simulado, el canal es variante en el tiempo, por esta razón la SNR medida no es fija.

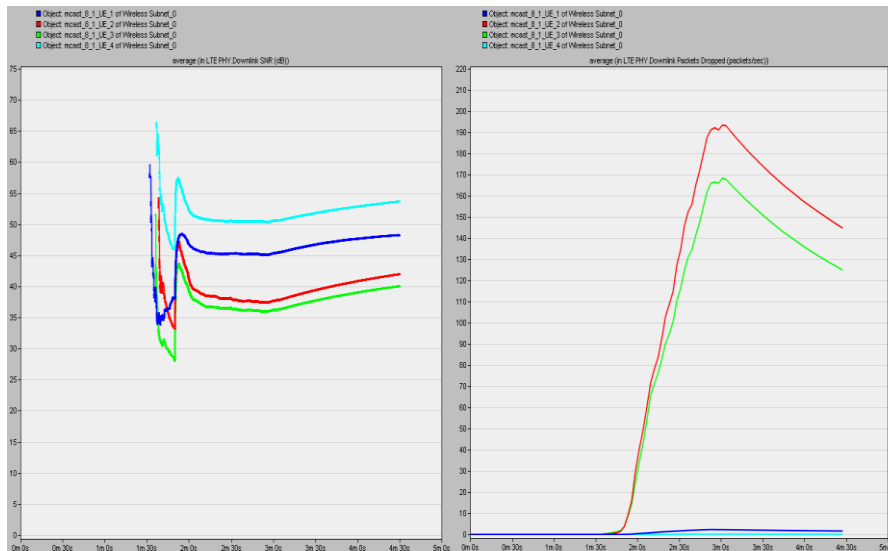


Figura 11. SNR (izquierda) y paquetes perdidos (derecha) en los diferentes usuarios

A pesar de que la distancia con la estación base afecta por igual a usuarios que utilizan servicios unicast y multicast, los recursos de la red LTE asignados para comunicaciones *multicast* son estáticos. Sin embargo, los recursos libres no utilizados en dicha comunicación, se acomodan dinámicamente sobre las comunicaciones punto a punto existentes.

De esta forma, dos usuarios con servicios *unicast* y *multicast* pueden tener la misma SNR pero no las mismas pérdidas. Esto se debe a que en servicios *multicast* los recursos asignados son constantes en toda la sesión, mientras que en servicios *unicast*, los recursos varían dinámicamente en función de los recursos no utilizados por el *multicast* y el número de usuarios *unicast* que compiten por dichos recursos.

Debido a esto, se pueden experimentar dos fenómenos: en el primero, la red está congestionada y tiene más dificultad para entregar contenido, que se estaba transmitiendo en MBMS, a través de una nueva comunicación *unicast* HTTP; en el segundo caso, la red dispone de capacidad suficiente para acomodar una nueva comunicación *unicast* con menor pérdida de información que la que se estaba experimentando en el servicio *multicast* que se está ejecutando. Uno de los parámetros que gobierna este comportamiento es el esquema de modulación y codificación MCS, fijo para MBMS y dinámico para las portadoras EPS.

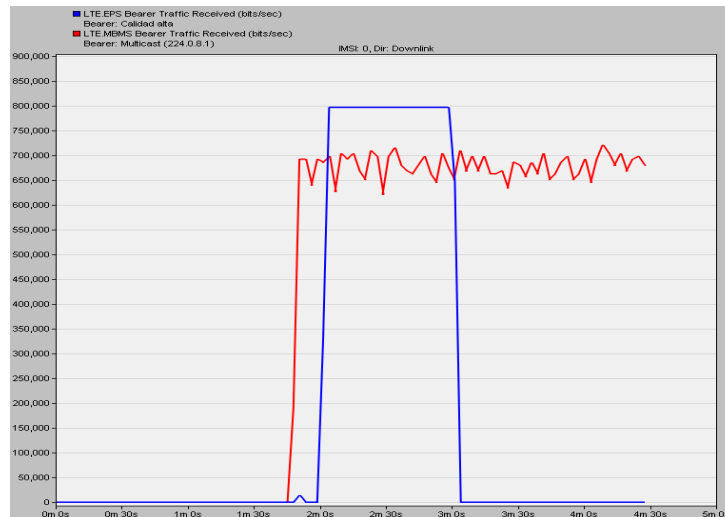


Figura 12. Escenario sin congestión

En la Figura 12 se muestra un usuario, situado a una determinada distancia del eNB, que ejecuta servicios *unicast* y *multicast* simultáneamente. Como se puede observar, la degradación de SNR afecta al servicio *multicast* (en rojo), puesto que los recursos que se le asignaron han permanecido fijos durante toda la sesión. Lo contrario ocurre en el servicio *unicast* (en azul), que no ha experimentado una degradación de la calidad del servicio debido a que la red dispone de capacidad suficiente para acomodar la comunicación con éxito.

En la Figura 13 se muestra el caso contrario. Por un lado, se observa que la capacidad disponible para la comunicación *multicast* se mantiene constante durante la sesión MBMS. Por otro, se observa que la red asigna dinámicamente la capacidad restante entre el conjunto de *usuarios unicast* que compiten por dichos recursos. De esto se deduce que, la conmutación a un servicio *unicast*, puede producirse en un momento de congestión inadecuado para atender dicha petición.

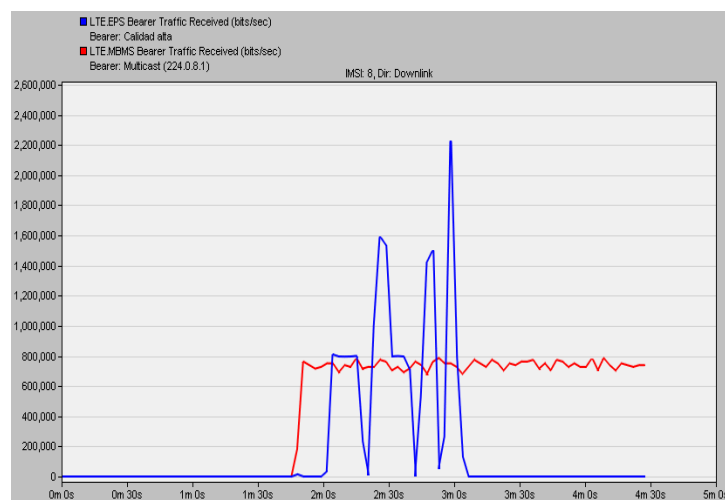


Figura 13. Escenario con congestión

A lo largo de este capítulo se ha presentado la configuración de una red LTE con un escenario de simulación sencillo. Para ello, se ha asignado el valor de diversos parámetros en diferentes niveles de la torre de protocolos y en múltiples elementos de la arquitectura de la red LTE.

Validado el escenario, se ha estudiado parte de la arquitectura definida en el Capítulo 3. En particular, la transmisión con pérdidas en flujos de video unicast y multicast sobre LTE. Con los resultados obtenidos, se puede concluir que el éxito de la integración de servicios *unicast* y *multicast* depende de varios parámetros de influencia. Las relaciones detectadas son:

- Relación entre paquetes perdidos y calidad del canal del usuario.
- Relación entre paquetes perdidos y *code rate* elegido para el FEC.
- Relación entre la sobrecarga de la cabecera FEC y el ancho de banda útil para la transmisión de video.
- Asignación de recursos y densidad de usuarios. Reserva dinámica de recursos para comunicaciones *unicast*, función de la densidad de usuarios, y reserva estática en comunicaciones *multicast*.

Para finalizar, me gustaría destacar que para realizar las diferentes pruebas de simulación se han aprovechado algunas funcionalidades avanzadas de la herramienta. OPNET Modeler cubre algunos aspectos que facilitan la simulación de proyectos complejos. La simulación de las redes LTE es un ejemplo de ello, el número de parámetros a configurar en escenarios de este tipo es elevado, por lo que se exige una metodología que mejore la experiencia del usuario y reduzca los tiempos de configuración. Esta metodología se ha basado en la importación y exportación de topologías con ficheros XML. En el futuro se propone una metodología de simulación paramétrica distribuida de cara a minimizar el tiempo de ejecución y hacer un mejor uso de los recursos computacionales disponibles.

Capítulo 5

Impacto del uso de técnicas FEC en servicios de video streaming multicast sobre LTE

Como se vio en el Capítulo 3, un uso más eficiente de los recursos radio es posible gracias al servicio punto-a-multipunto MBMS. Por otra parte, con MBSFN se evitan las interferencias destructivas en las áreas donde se produce un solapamiento de cobertura entre diferentes estaciones base. De esta forma, aumenta la relación señal a ruido.

Sin embargo, el uso de estas técnicas es incompatible con el ajuste dinámico de algunos parámetros de transmisión, como por ejemplo, el MCS. Es por ello que las transmisiones *unicast* se adaptan a la capacidad de la red existente, mientras que el soporte de los servicios *multicast* requiere establecer el MCS en capas superiores. Este problema ya fue discutido en el Capítulo 4.

La elección de un MCS, marca un cierto nivel de protección frente a errores. Para MCS más elevados se eligen constelaciones menos robustas a consta de: obtener una mayor tasa de servicio, y añadir más sobrecarga FEC a nivel físico (PHY-FEC).

El canal radio es variante con respecto los múltiples usuarios que reciben el servicios *multicast*. Para aumentar la robustez y fiabilidad de las transmisiones, 3GPP ha estandarizado el uso de códigos Raptor como técnica de protección AL-FEC sobre MBMS. En el caso de que el uso conjunto de las técnicas PHY-FEC y AL-FEC sea insuficiente para proteger un determinado segmento de video, se recuperará el mismo a través de una conexión HTTP *unicast*, lo que se discutió en la arquitectura propuesta del Capítulo 3.

En el presente capítulo, se analiza cómo debe configurarse el sistema LTE, así como los mecanismos de corrección de errores PHY-FEC y AL-FEC para maximizar la tasa de servicio al tiempo que se minimiza el número de retransmisiones *unicast*. Con este análisis, somos capaces de establecer la configuración apropiada para el despliegue de un servicio de video streaming *multicast* sobre LTE.

5.1 Modelo del sistema estudiado

La problemática expuesta exige la simulación y análisis de múltiples escenarios:

- Escenarios con diferente ISD entre los eNBs desplegados.
- Escenarios con diferente número de eNBs en el área MBSFN.
- Análisis utilizando diferente MCS (protección PHY-FEC).
- Análisis utilizando diferentes tasas de codificación en AL-FEC.
- Análisis utilizando diferentes períodos de protección del bloque FEC.
- Compromiso obtenido para maximizar la capacidad y la cobertura del servicio multicast.

Para la configuración de estos escenarios se han fijado numerosos parámetros, algunos comunes en todos los casos de estudio. Por ejemplo, en todos se ha definido un servicio de video streaming *multicast* con 399 receptores estáticos distribuidos uniformemente sobre el área MBSFN. En todos, se ha añadido un anillo exterior de estaciones base interferentes operando en la misma frecuencia. En la Tabla 4 se muestran los valores que adquieren algunos de los parámetros más significativos de las simulaciones realizadas.

Tabla 4. Parámetros del sistema

Parameter	Value
MBSFN area size	1 / 7 / 19 eNodeBs
Interference model	1 tier
eNodeBs geographical overlay	Hexagonal
ISD	500 m / 1732 m
Transmission power	45 dBm
Subframes reserved for MBMS	1
Cyclic prefix	Extended (16.7 μ s)
Bandwidth	10 MHz
Downlink base frequency	2110 MHz
Pathloss model	3GPP Urban Macrocell
Multipath channel model	ITU Pedestrian B
eNodeBs transmission antennas	1
Total number of UEs	399
UEs distribution	Uniform distribution
Modulation and Coding Schemes	MCS 4 / 5 / 6 / 7 / 8 / 10 / 15
Length of Raptor Codes segments	1 sec / 2 sec / 10 sec / 20 seg

En la Figura 14 se muestra uno de estos escenarios. En particular, el formado por siete estaciones base en el área MBSFN.

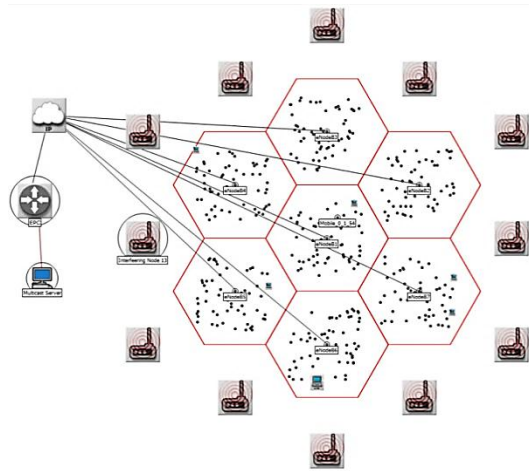


Figura 14. Escenario para un área MBSFN de 7 celdas

La Figura 15 muestra el sistema completo. El bloque *Transmission channel* representa la configuración a nivel físico del sistema LTE. Es decir, para la simulación de los escenarios es necesario especificar el modelo de propagación, la potencia de transmisión de las estaciones base, un modelo de interferencia, el área MBSFN y la distribución de los usuarios en dicho área. El bloque *Extracted physic parameters* muestra los parámetros físicos de influencia que son objeto de la investigación. Por un lado, el nivel de señal a ruido e interferencia (SINR) medido en cada usuario, parámetro depende de varios factores, como por ejemplo, la localización del usuario en la red, el área de cobertura, o el modelo de interferencia. A partir del SINR se obtienen los valores de BLER para cada usuario, donde la BLER es la probabilidad de error de bloque. Por otro lado, se varía el MCS utilizado para la transmisión *multicast*. Recuérdese que valores más altos de MCS dan una menor robustez ante errores y ofrecen una tasa mayor.

A nivel de red y transporte (bloque *Networking Level*) se muestra el uso del protocolo no fiable UDP, IP-Multicast y otros niveles inferiores de la torre de protocolos de LTE encargados de segmentar y concatenar datagramas IP en bloques de transporte físicos o *transport block*. Con el objeto de hacer coincidir la probabilidad de error de paquete (PER) con la BLER, se ha definido el tráfico para hacer coincidir el tamaño del paquete con el del *transport block*.

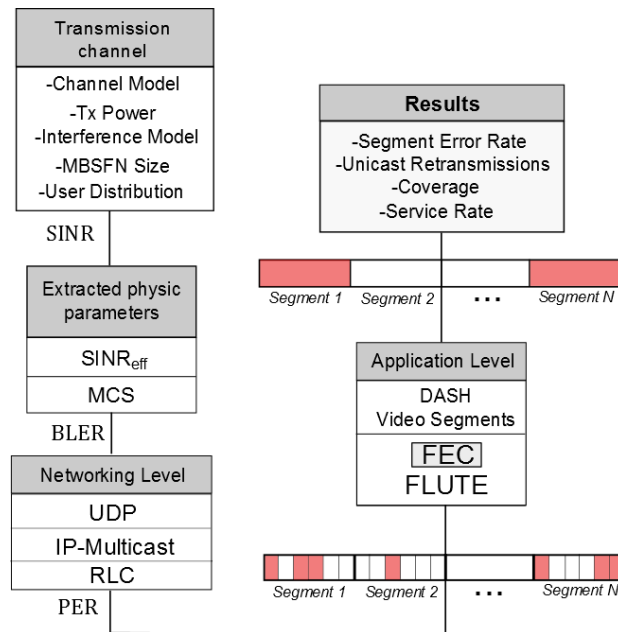


Figura 15. Arquitectura para simular el servicio de video streaming.

El nivel de aplicación (*Application Level*) ilustra los mecanismos que soportan el servicio de video streaming. Los segmentos de video DASH son transmitidos como objetos del protocolo FLUTE. Para ello, el video es pre-codificado y dividido en segmentos de una determinada duración. Dado que los servicios Live Streaming presentan requisitos de latencia estrictos (del orden de varios segundos), se recomienda el uso de segmentos de corta duración.

Por otra parte, cada segmento DASH es codificado de forma independiente utilizando el código Raptor. Mayores periodos de protección requerirían un menor número de símbolos de redundancia, sin embargo, para servicios de video, es necesario acotar retardos, como por ejemplo el zapping time. Si el tiempo de zapping, definido como el tiempo transcurrido entre el instante en el que usuario presiona un botón para cambiar de canal, hasta que el reproductor muestra el canal seleccionado, es demasiado grande, puede afectar a la calidad de servicio percibida por el usuario. Es por esta razón, que en el proceso de codificación, el periodo de protección debería fijarse al tamaño más pequeño posible.

En la misma figura, se muestra la transmisión de cada segmento de video, donde se envían tanto los k source symbols, como los r repair symbols. Se asume que un segmento se ha decodificado correctamente si la probabilidad de fallo en el proceso de decodificación es igual o menor al 1%.

Dado que el RFC de Raptor recomienda que cada símbolo sea enviado en un paquete distinto, y dado que se ha configurado el escenario para que el tamaño del

paquete coincida con el *transport block*, es posible calcular la probabilidad de fallo del decodificador como:

$$P(f_{RC}) = \sum_{n=0}^{k+r} P(f_{RC}|n) \times P(N = n)$$

Donde $P(f_{RC}|n)$ es la probabilidad de fallo del decodificador dado que se reciben n *encoding symbols*, y donde $P(N = n)$ es la probabilidad de recibir correctamente n símbolos. $P(f_{RC}|n)$ se calcula como [31]:

$$P(f_{RC}|n) = \begin{cases} 1 & \text{if } n < k \\ 0.85 \times 0.567^{n-k} & \text{if } n \geq k \end{cases}$$

Por otra parte, la probabilidad de recibir n símbolos se modela usando una distribución binomial, dada como:

$$P(N = n) = \binom{k+r}{n} \times (1 - PER)^n \times (PER)^{(k+r-n)}$$

donde la PER es la probabilidad de error de símbolo, y que ha sido calculada teniendo en cuenta las asunciones anteriores. Como se ha enunciado anteriormente, $P(f_{RC})$ es la probabilidad de fallo en el proceso de decodificación de un determinado segmento de video, pero también la probabilidad de retransmitir dicho segmento a través de una conexión HTTP *unicast*.

5.2 Resultados

Para la evaluación del sistema se ha utilizado un tiempo simulado de 300 segundos usando tres tamaños de área MBSFN. En todos los escenarios, los 399 usuarios se distribuyen aleatoriamente sobre dicho área. Las configuraciones utilizadas son:

- Área MBSFN de 1 celda.
- Área MBSFN de 7 celdas.
- Área MBSFN de 19 celdas.

Es importante destacar que aumentamos el tamaño del área MBSFN pero no el número de usuarios, por lo que la densidad de usuarios para áreas mayores es menor y la distribución menos representativa. Sin embargo, aumentar el número de usuarios incrementa excesivamente el coste computacional requerido para las simulaciones. Por esta razón, se sugiere una futura estrategia que distribuya los usuarios en diferentes anillos perimetrales.

Adicionalmente, se han obtenido resultados para dos casos extremos de *Inter Site Distance* (ISD): 500 m y 1000 m. Este parámetro define la distancia entre eNBs en la red LTE.

5.2.1 Análisis de cobertura frente a AL-FEC code rate

En esta subsección se analiza el área de cobertura en función del code rate del AL-FEC. Este análisis se ha realizado para las configuraciones de 1,7 y 19 celdas de área MBSFN. La transmisión *multicast* se ha configurado con distintos MCS. La figura 16 muestra, para un área MBSFN de 7 celdas con un ISD de 500m, el porcentaje promedio de usuarios que pueden decodificar correctamente la información sin solicitar retransmisiones *unicast*. De ello se deduce que aumentando la robustez del MCS, y el número de símbolos de redundancia en el código Raptor (i.e., reduciendo el code rate), se alcanza un área de cobertura mayor.

Por otro lado, la figura 17 muestra, en función del code rate, la cobertura de los usuarios para las distintas áreas MBSFN configuradas. Se ha representado el caso que presenta una mayor diferencia, en este caso, para MCS 15. De esta figura, se deduce que existe una mejora cualitativa al pasar de usar una estación base, a usar un área MBSFN de 7 celdas. Sin embargo, aumentar a 19 el número de celdas del área MBSFN no presenta una gran ventaja, y sin embargo, aumenta la complejidad de coordinar un número mayor de estaciones base.

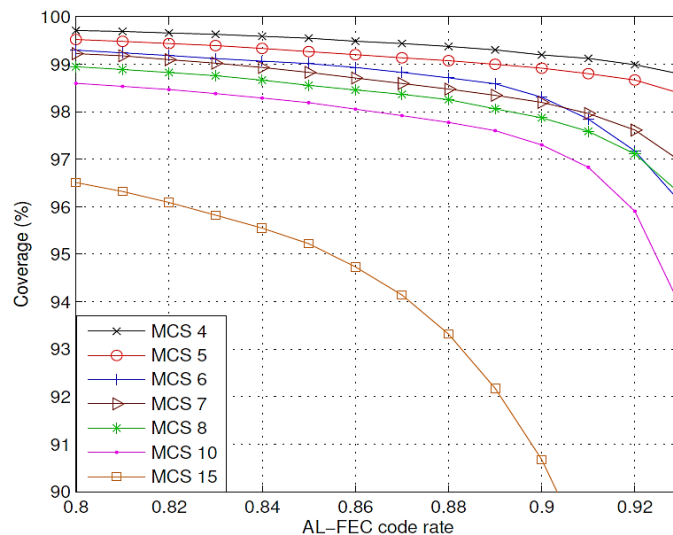


Figura 16. Cobertura Vs AL-FEC code rate para un área MBSFN de 7 celdas

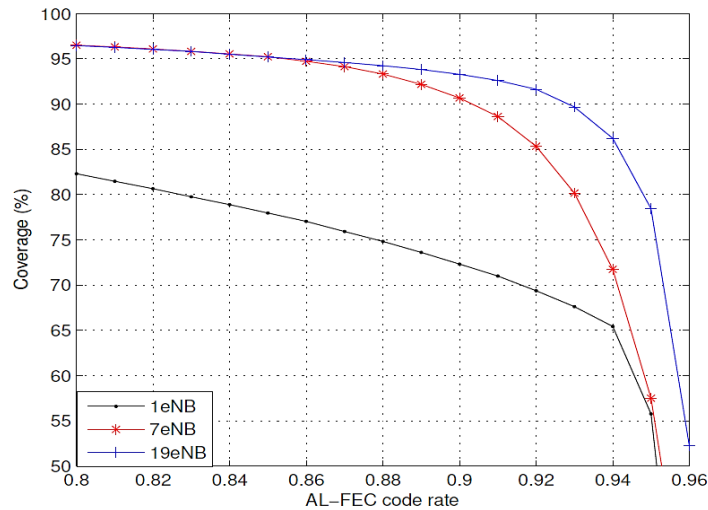


Figura 17. Cobertura Vs AL-FEC code rate para MCS 15

5.2.2 Análisis de tasa de servicio frente a cobertura

El uso de AL-FEC incrementa la robustez y fiabilidad del servicio. Sin embargo, introducir símbolos de redundancia afecta a la tasa máxima de datos del servicio. Este análisis se centra en estudiar la tasa de servicio en función del área de cobertura que puede cubrirse.

La Figura 18 muestra, para un área MBSFN con 7 celdas, un ISD de 500m y diferentes MCS, la tasa de datos del servicio en función de la cobertura. Se puede observar que la utilización de MCS menos robustos proporciona tasas mayores para niveles de cobertura menores. Por lo tanto, para reducir el número de retransmisiones, se deben seleccionar esquemas de modulación y codificación más robustos con tasa de servicio menores. En este punto, se debe resaltar que las tasas obtenidas se corresponden con la reserva de una única subtrama por trama radio LTE. Según el estándar, sería posible reservar hasta un máximo de 6 subtramas por trama en servicios MBMS.

La Figura 19 muestra, para un MCS fijo de 15 y para las distintas áreas MBSFN, la tasa de datos del servicio en función de la cobertura. Al igual que en el análisis anterior, el uso de un área MBSFN de 7 celdas aumenta apreciablemente la tasa con respecto a una estación base. Sin embargo, el uso de 19 celdas en el área MBSFN no muestra una ventaja suficiente como para despreciar el coste de sincronización de un área MBSFN mayor.

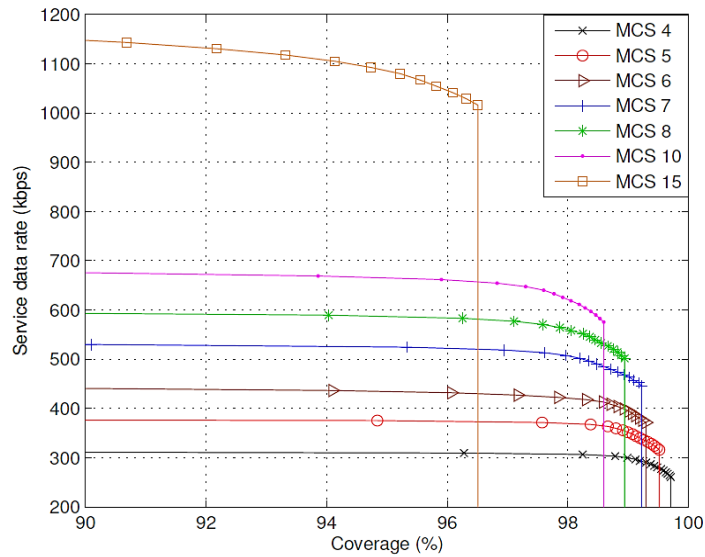


Figura 18. Tasa de servicio de datos Vs Cobertura en área MBSFN de 7 celdas.

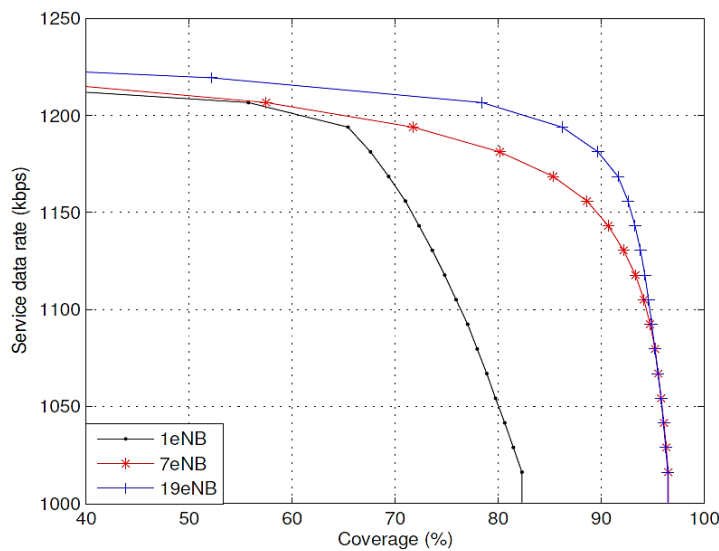


Figura 19. Tasa de datos Vs Cobertura para MCS 15

5.2.3 Análisis del ISD en el despliegue del sistema LTE

Uno de los factores que impactan directamente en las condiciones del canal radio de los usuarios, es la distancia entre estaciones base o ISD. Este parámetro fija la influencia del tamaño del área de la celda sobre la cobertura del servicio *multicast*. La Figura 20 muestra, para: ISDs de 500 y 1732 metros, área MBSFN de 7 celdas y distintos MCS, la tasa de servicio alcanzada en función de la cobertura. Es destacable el pobre rendimiento obtenido para un ISD de 1732 metros. Esto muestra la importancia de establecer un tamaño apropiado de las celdas, para desplegar servicios LTE sobre entornos urbanos.

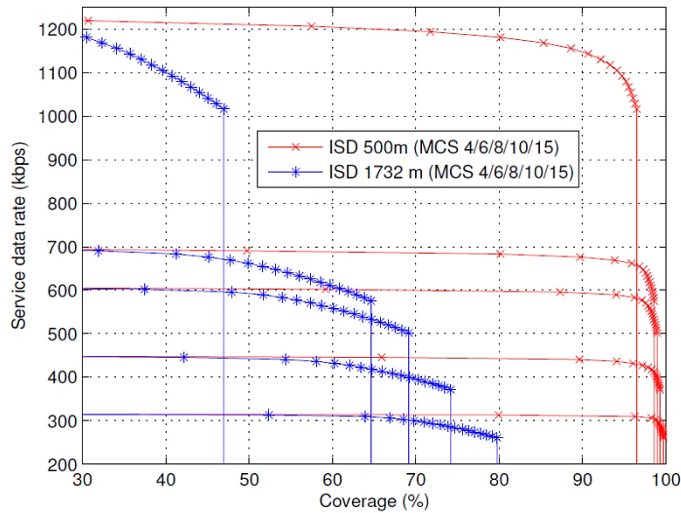


Figura 20. Tasa de datos Vs Cobertura para distintos ISDs

5.2.4 Análisis de la duración de los segmentos de video protegidos por AL-FEC

Los servicios LiveStreaming son intolerantes a retardos. Por esta razón, se recomiendan segmentos de video entre 2 y 10 segundos de duración. El análisis realizado en esta subsección varía el tamaño del segmento para observar cómo afecta a la tasa de servicio y cobertura. La Figura 21 muestra, para un área MBSFN de 7 celdas con un ISD de 500 metros y MCS 15, la tasa máxima obtenida para diferentes periodos de protección y en función de la cobertura. De esta figura, se puede observar que el incremento de tasa que se obtiene al usar periodos de protección mayores, no justifica el aumento de retardo (zapping time) asociado a dicho incremento.

Por otra parte, se observa que aumentar la duración del segmento de video por encima de 20 segundos no aporta una mejora significativa en la tasa de servicio.

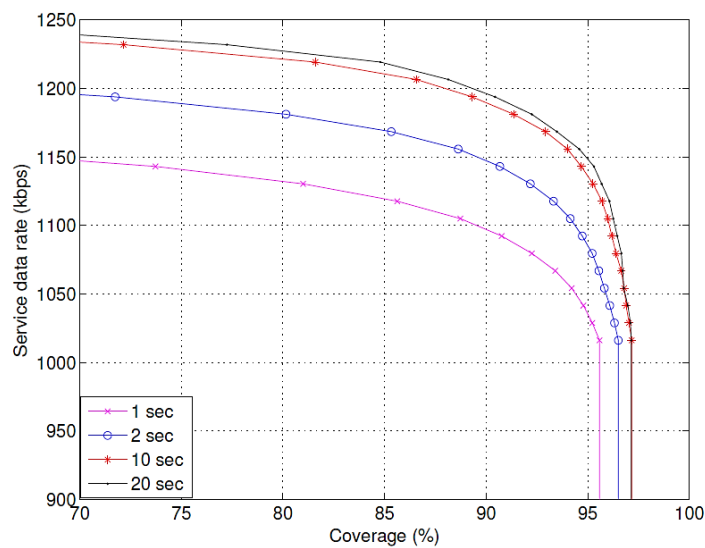


Figura 21. Tasa de servicio Vs Cobertura para diferentes valores del periodo de protección

5.2.5 Compromiso AL-FEC Vs PHY-FEC para maximizar la tasa de datos del servicio.

Los resultados finales de este estudio se centran en la integración de las técnicas AL-FEC y PHY-FEC para maximizar la tasa de datos del servicio. Para analizar el compromiso o *trade-off* entre ambas técnicas, se ha evaluado la tasa máxima de datos en función de la cobertura en un escenario de ISD 500 metros. Este análisis se ha llevado a cabo para diferentes áreas de MBSFN, observándose, que para el área de 7 celdas, la tasa de servicio obtenida entre el 90% y 96% de cobertura, es casi dos veces la de un área de una única estación base. Como se puede intuir de los análisis anteriores, y como puede observarse también en la Figura 22, el rendimiento en áreas MBSFN de 7 y 19 celdas es prácticamente el mismo.

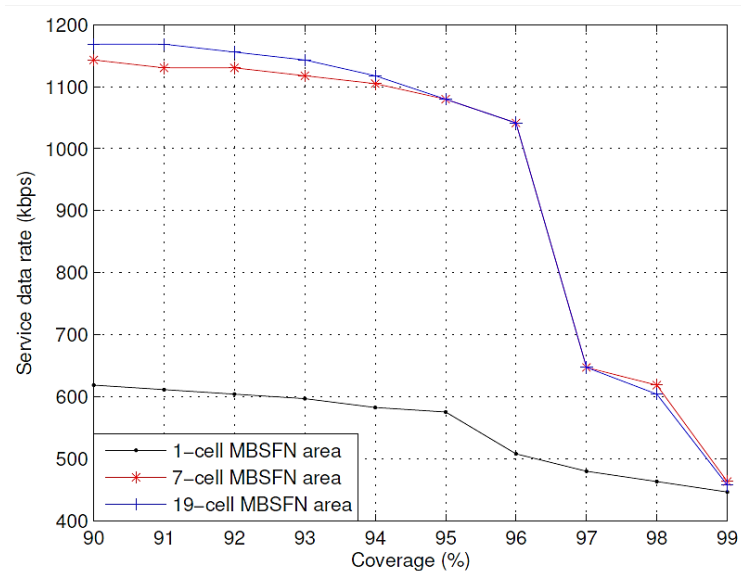


Figura 22. Tasa de datos máxima Vs Cobertura para AL-FEC/PHY-FEC trade-off

Capítulo 6

Conclusiones y líneas futuras

6.1 Conclusiones

A lo largo del presente Trabajo Fin de Master, se obtienen un conjunto de conclusiones que van a ser descritas en este capítulo.

El trabajo comienza con la revisión del estado del arte, de donde se concluye que es conveniente y necesario utilizar redes de distribución de contenidos CDN, o nuevos esquemas híbridos CDN-P2P para optimizar el problema de distribución. Se estudian diversas tecnologías para ofrecer servicios de *streaming*, y se concluye, que para comunicaciones *unicast*, las técnicas de streaming adaptativo son especialmente relevantes de cara a solucionar algunos problemas percibidos por los usuarios en la reproducción del video, como por ejemplo, frecuentes interrupciones o paradas en la reproducción. En este punto, se destaca DASH como solución estándar.

También se revisan tecnologías más adecuadas para comunicaciones punto-a-multipunto sobre canales no fiables, de lo que se concluye que es necesario utilizar técnicas de corrección de errores hacia delante o en el destino FEC para proteger la información enviada sobre protocolos no fiables como UDP. Del estudio de las diversas técnicas de corrección destacan los códigos Raptor (estándar del 3GPP como código AL-FEC sobre MBMS), pero también nuevas propuestas que mejoran el rendimiento del estándar, códigos RaptorQ y RS+LDPC-Staircase.

En el Capítulo 3 se presenta una arquitectura para ofrecer servicios de video streaming sobre LTE. Por un lado, se selecciona un esquema híbrido CDN-P2P que es congruente con las limitaciones de ancho de banda existentes en las redes móviles. Por otro, se presenta una arquitectura que realiza el envío *multicast* sobre FLUTE y MBMS, y la recuperación *unicast* en el caso de que las técnicas FEC se muestren incapaces de recuperar los segmentos de video.

En el Capítulo 4 se analiza parte de la arquitectura anterior. Tras las múltiples simulaciones realizadas, se detectan varios parámetros de influencia que afectan directamente en las retransmisiones *unicast*: SINR percibido por cada usuario, tasa de codificación o code rate en AL-FEC ($BW_U = BW_{Total} - R$), esquema de modulación y codificación (MCS) y asignación o reserva de recursos.

Con el objetivo de minimizar el número de retransmisiones y de maximizar la tasa de servicio, se realiza, en el Capítulo 5, un análisis sobre el impacto del uso de técnicas FEC en servicios de video streaming *multicast* sobre LTE. Con este propósito, se desarrolla una herramienta que permite estudiar la interrelación de AL-FEC y PHY-FEC en diferentes escenarios, con el fin de ayudar a un operador de red a desplegar un servicio de video streaming sobre LTE.

En el Capítulo 5 se presentan los resultados obtenidos para un escenario concreto. En ellos se observa:

- La relación entre MCS y redundancia en AL-FEC con la cobertura y tasa de datos del servicio.
- El impacto de utilizar áreas MBSFN. En concreto, y para este escenario, se observa que un área MBSFN de 7 eNB mejora el rendimiento ofrecido por una estación base. Por otro lado, se observa que la mejora utilizando 19 eNBs es pequeña.
- La influencia de la longitud del periodo de protección en la tasa de datos de servicio. En este escenario, se recomienda el uso de periodos de protección pequeños, puesto que la mejora en términos de tasa no es significativa, y sin embargo, aumentar el periodo de protección aumenta el zapping time.
- La posibilidad de alcanzar mayores coberturas y tasas de datos utilizando ISDs menores.
- El compromiso entre AL-FEC y PHY-FEC para maximizar la tasa de datos y minimizar el número de retransmisiones totales. En el escenario definido, es posible duplicar la tasa de datos del servicio multicast utilizando un área MBSFN de 7 celdas, y hasta una cobertura del servicio del 96%.

En definitiva, la herramienta diseñada nos permite estudiar múltiples parámetros que deben ser considerados para proveer servicios de video streaming multicast sobre LTE.

6.2 Líneas Futuras

Como línea futura se propone replicar el estudio realizado en el capítulo 5 con nuevas propuestas existentes, como por ejemplo, RaptorQ y RS+LDPC-Staircase. Este análisis compararía el actual estándar con estos nuevos códigos y establecería nuevos valores de los parámetros analizados en el capítulo 5 para el despliegue del servicio de video streaming *multicast*.

El sistema simulado no tiene en cuenta otro posible tráfico existente en la red. Las simulaciones solo consideran tráfico *multicast*, pero en la red existe tráfico *unicast* de muy diversa naturaleza: servicio de correo, FTP, VoD, voz, etcétera. Por esta razón, se propone estudiar el impacto de los servicios *multicast* (y sus retransmisiones) sobre el resto de servicios *unicast* ya existentes en la red.

Otra posible línea de investigación explotaría las características del sistema LTE-A para mejorar la calidad de la experiencia del usuario en el uso de servicios de video streaming *multicast*. Para ello se propone estudiar, como afecta que un usuario que está recibiendo el contenido desde el área MBSM, pase a recibir el contenido desde su propia femtocelda o HeNB.

Por último, y dado que se ha visto la importancia capital del uso de las redes de distribución de contenidos, se propone estudiar la QoS percibida por el usuario cuando se despliega una CDN conectada en el BM-SC (sistema LTE), y conectada al L-GW (sistema LTE-A).

Para completar este estudio se sugiere realizar encuestas de valoración subjetivas sobre una maqueta real que integre todos los elementos de la arquitectura vista en el capítulo 3.

Referencias

- [1] ISO/IEC 23009-1, "Information technology Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats", April 2012
- [2] 3GPP, TS 123 246 V8.4.0; Universal Mobile Telecommunications System (UMTS); LTE; Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description. (3GPP TS 23.246 version 8.4.0 Release 8), June 2009
- [3] IETF RFC 6726, "FLUTE - File Delivery over Unidirectional Transport", November 2012.
- [4] Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs, 3GPP TS 26.346 v11.6.0, Sep 2013.
- [5] OPNET. (2014). Opnet Modeler official page. [Online]. Available: <http://www.riverbed.com>
- [6] "2G, 3G, 4G(LTE) Services and Subscribers: Voice, SMS/MMS and Broadband Report". Infonetics, July 2012.
- [7] "Apps and the mobile Internet. The battle of the platforms: both native and Web apps Report". DigiWorld NEWS 585, January 2012.
- [8] Cisco Visual Networking Index, Global Mobile Data Traffic Forecast Update, 2011–2016.
- [9] Jan Coppens, Tim Wauters, Filip De Turck, Bart Dhoedt and Piet Demeester, "Design and Performance of a Self-Organizing Adaptive Content Distribution Network", 2006.
- [10] Jaison Paul Mulerikkal, Ibrahim Khalil, "An Architecture for Distributed Content Delivery Network", 2007
- [11] Faqir Zarrar Yousaf, Marco Liebsch, Andreas Maeder, and Stefan Schmid, "Mobile CDN Enhancements for QoE-Improved Content Delivery in Mobile Operator Networks", NEC Laboratories Europe, IEEE 2013
- [12] Stella Spagna, Marco Liebsch, Roberto Baldessari, Saverio Niccolini, and Stefan Schmid, Rosario Garroppo, Kazunori Ozawa and Jun Awano, "Design Principles of an Operator-Owned Highly Distributed Content Delivery Network", IEEE 2013
- [13] N. Amram et al., "QoE-Based Transport Optimization for Video Delivery over Next Generation Cellular Networks", 6th Wksp. Multimedia Applications over Wireless Networks, 2011.
- [14] Magharei, N.; Rejaie, R.; Yang Guo, "Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches," INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, vol., no., pp.1424,1432, 6-12 May 2007
- [15] Feng Wang, Yongqiang Xiong, Jiangchuan Liu, –mTreebone: a hybrid tree/mesh overlay for application layer live video multicast, IEEE ICDCS 2007, pp. 49-56.
- [16] ZhiHui Lu, Ye Wang, and Yang Richard Yang, "An Analysis and Comparison of CDN-P2P-hybrid Content Delivery System and Model". JCM, March 2012.

- [17] Lv, Zhi-Huz; Chen, Li-Jiang; Wu, Jie; Deng, Da; Huang, Si-Jia; Huang, Yi, "PROSE: Proactive, Selective CDN Participation for P2P Streaming", *Journal of Computer Science and Technology*, Volume 28, Number 3, May 2013 , pp. 540-552(13)
- [18] Jie Wu, ZhiHui lu, BiSheng Liu, ShiYong Zhang, —PeerCDN: a novel P2P network assisted streaming content delivery network scheme, *Proceeding of IEEE CIT2008*, IEEE Computer Society, pp. 601-606.
- [19] ZhiHui Lu Jie Wu Chuan Xiao WeiMing Fu YiPing Zhong, —WS-CDSP: a novel web services-based content delivery service peering scheme, *Proceeding of 2009 IEEE SCC 2009*, IEEE Computer Society, pp.348-355.
- [20] Bronzino, F.; Gaeta, R.; Grangetto, M.; Pau, G., "An adaptive hybrid CDN/P2P solution for Content Delivery Networks," *Visual Communications and Image Processing (VCIP)*, 2012 IEEE , vol., no., pp.1,6, 27-30 Nov. 2012
- [21] Hao Yin, Xuening Liu, Tongyu Zhan, Vyas Sekar, Hui Zhang,—Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky, *ACM Multimedia 2009*, pp.25-34.
- [22] Stefan Lederer, Christopher Müller, and Christian Timmerer, "Towards Peer-Assisted Dynamic Adaptive Streaming over HTTP", *Packet Video Workshop (PV)*, 2012 19th International, Mayo 2012
- [23] I. Unanue, I. Urteaga, R. Husemann, J. Del Ser, V. Roesler, A. Rodriguez, and P. Sanchez. A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance. In *Recent Advances on Video Coding*, pages 1--24. InTech, June 2011.
- [24] Christian Sieber, Tobias Hoßfeld, Thomas Zinner, Phuoc Tran-Gia and Christian Timmerer, "Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC", *IFIP/IEEE International Workshop on Quality of Experience Centric Management (QCMAN)*, Belgium, from May 27-31, 2013.
- [25] Thorsten Lohmar, Torbjörn Einarsson, Per Fröjdh, Frédéric Gabin and Markus Kampmann, "Dynamic Adaptive HTTP Streaming of Live Content", *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2011 IEEE International Symposium, Junio 2011
- [26] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, RaptorQ Forward Error Correction Scheme for Object Delivery, RFC 6330, IETF, Aug 2011.
- [27] V. Roca, M. Cunche, C. Thienot, J. Detchart, and J. Lacan, "RS + LDPC-Staircase Codes for the Erasure Channel: Standards, Usage and Performance," in *9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2013)*, Aug. 2013.
- [28] M. Luby, A. Shokrollahi, T. Stockhammer, Raptor Forward Error Correction Scheme for Object Delivery, RFC 5053, IETF, Oct 2007
- [29] Shiann-Tsong Sheu; Chi-Hao Huang, "Mixed P2P-CDN system for media streaming in mobile environment," *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2011 7th International , vol., no., pp.657,660, 4-8 July 2011

- [30] Zhe Li and Gwendal Simon, "In a Telco-CDN, Pushing Content Makes Sense" IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT 2013
- [31] "Technical Specification Group Services and System Aspects; MBMS; Protocols and codecs (Release 9)", 3GPP TS 26.346 V9.4.0, Sep 2010.