



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño y Desarrollo de un Sistema de Gestión Inteligente de QoE para Redes HD y Estereoscópicas IPTV

PROGRAMA DE DOCTORADO EN INFORMÁTICA

Autor: Alejandro Cánovas Solbes

Director/co-Director: Jaime Lloret Mauri

Jesús Tomás Gironés

Abril de 2016

Dedicatoria

A la persona que ilumina mi camino, mi fuente de inspiración, mi razón de ser y la que nunca deja de confiar en mí. Porque sin tí, los días serían de otro color, las noches más largas y mi corazón más pequeño. Recuerda, por siempre nuestro ...

Gracias Natalia Cots.

Resumen

Las conexiones de acceso a Internet de banda ancha permiten a los proveedores de servicios de Internet (Internet Service Provider (ISP)) ofrecer servicios a los hogares tales como datos, Voz sobre IP (VoIP), Televisión sobre IP (IPTV) y actualmente 3D-TV sobre IP (3D-IPTV). Es por esto que el número de proveedores de servicios de IPTV está aumentando considerablemente en los últimos años. Gracias a la evolución tanto a nivel de sistemas, como de redes de comunicación como de dispositivos, la entrega de este tipo de servicios es posible pero no siempre con las máximas garantías de calidad. Por este motivo, una de las principales cuestiones a tener en cuenta por parte del proveedor de servicios de IPTV es garantizar la calidad de experiencia (Quality of Experience (QoE)) percibida por el usuario final. Para conseguir este objetivo, en la siguiente tesis doctoral se propone un sistema de gestión inteligente basado en métodos inductivos de predicción para garantizar la QoE del usuario final. Uno de los aspectos importantes a tener en cuenta en el desarrollo del sistema de gestión es el incluir los parámetros que afectan a la QoE. Para ello, se analizarán aquellos parámetros que afecten a la degradación del flujo de vídeo recibido por el usuario final a través del servicio de IPTV. A nivel de red, se identificarán dichos parámetros como aquellos que afectan a la calidad de servicio (Quality of Service (QoS)) como son el jitter, el retardo, los paquetes perdidos y el ancho de banda principalmente. A nivel de usuario, estos parámetros afectan a la percepción subjetiva del usuario al visualizar el vídeo. Comprobamos como efectos derivados de la compresión, la cuantificación, el bitrate, etc, afectan también a dicha percepción.

Alejandro Cánovas Solbes

Abstract

Broadband Internet access connections allow Internet Service Providers (ISP) to offer several types of services to home customers such as data, Voice over IP (VoIP), Internet Protocol Television (IPTV) and now 3D Internet Protocol Television (3D-IPTV). That is why the number of IPTV service providers is increasing considerably in recent years. Thanks to the evolution at many levels of the communication systems, communication networks and devices, to deliver these services is possible, but the maximum quality is not always guaranteed. For this reason, one of the main issues to be considered by the IPTV service providers is to guarantee the Quality of Experience (QoE) perceived by the end user. In order to achieve this goal, in this PhD Thesis we propose an intelligent management system based on inductive prediction methods to guarantee the QoE of the end-user. One of the important aspects to be considered in the development of the management system is to include all the parameters that affect the QoE. With this purpose, we will analyze the parameters that affect the degradation of the video stream received by the end user through the IPTV service. At the network level, we will identify the main parameters which affect the Quality of Service (QoS), such as jitter, delay, lost packets and bandwidth. At the user level, these parameters affect to the subjective perception of the user when watching the video. We also checked that effects derived from the compression, quantization, and bitrate affect this perception too.

Alejandro Cánovas Solbes

Resum

Les connexions d'accés a Internet de banda ampla permeten als Proveïdors de Servicis d'Internet (ISP) oferir servicis a les llars com ara dades, veu sobre IP (VoIP), Televisió sobre IP (IPTV) i actualment 3D-TV sobre IP (3D-IPTV). És per açò que el nombre de proveïdors de servicis d'IPTV està augmentant considerablement en els últims anys. Gràcies a l'evolució tant a nivell de sistemes, com de xarxes de comunicació com de dispositius, l'entrega d'este tipus de servicis és possible però no sempre amb les màximes garanties de qualitat. Per este motiu, una de les principals qüestions a tindre en compte per part del proveïdor de servicis d'IPTV és garantir la qualitat d'experiència (Quality of Experience, QoE) percebuda per l'usuari final. Per a aconseguir este objectiu, en la següent tesi doctoral es proposa un sistema de gestió intel·ligent basat en mètodes inductius de predicció per a garantir la QoE de l'usuari final. Un dels aspectes importants a tindre en compte en el desenvolupament del sistema de gestió es incloure els paràmetres que afecten la QoE. Per a això, s'analitzaran aquells paràmetres que afecten la degradació del flux de vídeo rebut per l'usuari final a través del servici d'IPTV. A nivell de xarxa, s'identificaran dits paràmetres com aquells que afecten la qualitat de Servicis (Quality of Service, QoS) com són el jitter, el retard, els paquets perduts i l'ample de banda principalment. A nivell d'usuari, estos paràmetres afecten la percepció subjectiva de l'usuari al visualitzar el vídeo. Comprovem com efectes derivats de la compressió, la quantificació, el bitrate, etc, afecten també a dita percepció.

Alejandro Cánovas Solbes

Índice general

Dedicatoria	III
Resumen	V
Abstract	VII
Resum	IX
Índice general	XI
Lista de figuras	XV
Lista de tablas	XXI
1 Introducción	1
1.1 Introducción	1
1.2 Objetivos y motivación	3
1.3 Precedentes	5
1.4 Estructura de la memoria	6
2 Estado del arte	9
2.1 Introducción	9
2.2 Arquitectura de red para IPTV.	12
2.3 Proceso de transcodificación en IPTV	15

2.4 Algoritmos de gestión	16
2.5 Sistemas de gestión	17
3 Conceptos generales	21
3.1 Introducción a QoE y QoS.	21
3.2 VQ y MOS	24
3.3 Métodos subjetivos, objetivos e híbridos para el cálculo de la VQ	27
3.3.1 Métodos subjetivos	28
3.3.2 Métodos objetivos	29
3.3.3 Métodos híbridos	31
3.4 Parametrización general de QoE y QoS	32
3.4.1 Parámetros de medida de la QoS	32
3.4.2 Parámetros de QoE en transmisiones de vídeo 2D	35
3.4.3 Parámetros de QoE en transmisiones de vídeo 3D/estereoscópica	37
3.5 Infraestructura general de red para IPTV	39
3.5.1 Arquitectura de red para IPTV	39
3.6 Proceso para la solicitud de un servicio de IPTV	42
3.7 Algoritmo de gestión global	43
3.7.1 Activación de un evento de petición de servicio IPTV.	44
3.7.2 Activación de un evento de alarma	47
3.8 Proceso general de análisis de la QoE.	49
3.9 Banco de pruebas para el estudio de la QoE	50
3.9.1 Configuración para la transmisión de vídeo utilizando VLC/Evalvid.	51
3.9.2 Configuración de la red de distribución	54
3.10 Teoría de colas, prioridad de servicios y cambios en la red	54
3.11 Conclusiones	56
4 Medidas preliminares	59
4.1 Roaming y Handover.	59
4.1.1 Proceso de Handover en base a la RSSI	60
4.1.2 Caso de estudio: análisis del proceso de handover durante una transmisión de vídeo	63
4.2 Compresión y contenedores	66
4.2.1 Estándares de compresión: familia MPEG y serie ITU H-XXX	67
4.2.2 Formatos contenedores multimedia	72

4.2.3	Códecs de compresión y contenedores para móviles	73
4.2.4	Códecs de compresión 3D/ST	76
4.3	Cuantificación	79
4.3.1	Caso de estudio: cómo afecta el Q_s a la VQ.	80
4.3.2	Caso de estudio: cómo afecta el Q_s a las condiciones de red.	82
4.4	Transcodificación	86
4.4.1	Caso de estudio: mejora de un servicio de IPTV con límites de BW utilizando transcodificación	89
4.5	Conclusiones	93
5	Sistema Inteligente de gestión	95
5.1	Introducción	95
5.2	Desarrollo teórico de los sistemas de predicción para el sistema de gestión	97
5.2.1	Estimación basada en Redes neuronales de unidades producto	97
5.2.2	Estimación basada en maquinas de soporte vectorial para regresión	101
5.3	Sistema de gestión basado en métodos de predicción.	109
5.4	Conclusiones	110
6	Sistemas de gestión basados en QoE en entornos reales de red	111
6.1	Sistema de gestión en entornos cableados de red	111
6.1.1	Introducción	111
6.1.2	Algoritmo de gestión.	112
6.1.3	Análisis de los parámetros de QoS que influyen en la QoE.	116
6.1.4	Conclusiones.	119
6.2	Sistema de gestión en entornos inalámbricos de red	120
6.2.1	Introducción	120
6.2.2	Algoritmo de gestión del servidor IPTV	121
6.2.3	Algoritmo del cliente IPTV.	123
6.2.4	Conclusiones.	123
6.3	Sistema de gestión para dispositivos de telefonía móvil	124
6.3.1	Introducción	124
6.3.2	Algoritmo de gestión de dispositivos móviles	125
6.3.3	Protocolo de red	126
6.3.4	Análisis de las características de los dispositivos móviles	128
6.3.5	Análisis de los parámetros de red y sistemas de codificación en entornos móviles.	129

6.3.6 Conclusiones	136
6.4 Sistema de gestión basado en métodos inteligentes y su estudio en la transmisión de vídeo estereoscópico	137
6.4.1 Introducción	137
6.4.2 Algoritmo de gestión para la mejora en la entrega de vídeo 3D/estéreo . . .	137
6.4.3 Pre-procesado de datos, extracción de características y etiquetado	139
6.4.4 Aprendizaje y test de los sistemas de predicción	148
6.4.5 Conclusiones	154
6.5 Conclusiones	155
7 Software de gestión	157
7.1 Introducción	157
7.2 Estructura del software	158
7.2.1 Pantalla de test	158
7.2.2 Pantalla de monitorización y gestión	162
7.3 Descripción del funcionamiento del software	169
7.4 Integración del software de gestión	174
7.5 Conclusiones	178
8 Conclusion	179
8.1 Achievement of objectives	179
8.2 Conclusion of the PhD Thesis	180
8.3 Personal contributions	181
8.4 Publications derived from the research work	181
8.5 Future research lines	182
Glosario	185
Referencias	191

Índice de figuras

3.1. Comparación entre el vídeo emitido y recibido para obtener el MOS	26
3.2. Tiempo de zapping para diferentes redes de acceso.	35
3.3. Errores perceptibles a nivel de imagen que afectan a la QoE	36
3.4. Arquitectura general de red para servicios de IPTV	40
3.5. Algoritmo de petición de servicio IPTV	43
3.6. Algoritmo de gestión global para servicios de IPTV	45
3.7. Algoritmo simplificado del sistema global de gestión.	48
3.8. Protocolo para la petición de un servicio de IPTV.	49
3.9. Protocolo para la activación de una alarma en el controlador de vídeo.	49
3.10. Proceso de QoE para el control de la degradación de la QoE. . . .	49
3.11. Arquitectura del banco de pruebas.	50
4.1. Proceso de handover de un cliente IPTV en un entorno inalámbrico.	61
4.2. Protocolo de acceso a un servicio de IPTV en redes inalámbricas .	62
4.3. Proceso de handover de un cliente IPTV en un entorno inalámbrico.	64
4.4. Número de paquetes recibidos por segundo.	65
4.5. Jitter por muestra.	65
4.6. Retardo por muestra.	66
4.7. Número de paquetes perdidos en % por muestra.	66

4.8. Ancho de banda utilizado por diferentes códecs durante la transmisión de un flujo de vídeo.	71
4.9. Ancho de banda medio utilizado por cada uno de los códecs.	71
4.10. Comparación del tiempo de transcodificación.	75
4.11. Tiempo para comprimir un segundo de vídeo.	75
4.12. Comparación del ancho de banda consumido.	76
4.13. Comparación de la tasa de compresión.	76
4.14. Imagen extraída de un vídeo con formato 2D+Z.	77
4.15. Imagen extraída de un vídeo con formato estéreo.	78
4.16. Calidad de vídeo objetiva en recepción para diferentes QS en vídeos Mpeg2.	81
4.17. Calidad de vídeo objetiva en recepción para diferentes QS en vídeos Mpeg4.	81
4.18. Proceso de codificación y ubicación de los dispositivos IPTV	82
4.19. Ancho de banda de una transmisión de vídeo Mpeg2 a diferentes QS.	83
4.20. Ancho de banda de una transmisión de vídeo Mpeg4 a diferentes QS.	83
4.21. Retardo cuando el ratio de pérdidas de paquetes es del 0.1 % para diferentes QS en vídeos Mpeg2.	84
4.22. Retardo cuando el ratio de pérdidas de paquetes es del 1 % para diferentes QS en vídeos Mpeg2.	84
4.23. Retardo cuando el ratio de pérdidas de paquetes es del 3 % para diferentes QS en vídeos Mpeg2.	85
4.24. Retardo cuando el ratio de pérdidas de paquetes es del 0.1 % para diferentes QS en vídeos Mpeg4.	85
4.25. Retardo cuando el ratio de pérdidas de paquetes es del 1 % para diferentes QS en vídeos Mpeg4.	86
4.26. Retardo cuando el ratio de pérdidas de paquetes es del 3 % para diferentes QS en vídeos Mpeg4.	86
4.27. Jitter cuando el ratio de pérdidas de paquetes es del 0.1 % para diferentes QS en vídeos Mpeg2.	87

4.28. Jitter cuando el ratio de pérdidas de paquetes es del 1 % para diferentes QS en videos Mpeg2.	87
4.29. Jitter cuando el ratio de pérdidas de paquetes es del 3 % para diferentes QS en videos Mpeg2.	87
4.30. Jitter cuando el ratio de pérdidas de paquetes es del 0.1 % para diferentes QS en videos Mpeg4.	87
4.31. Jitter cuando el ratio de pérdidas de paquetes es del 1 % para diferentes QS en videos Mpeg4.	88
4.32. Jitter cuando el ratio de pérdidas de paquetes es del 3 % para diferentes QS en videos Mpeg4.	88
4.33. Conversion del formato utilizando un transcodificador de vídeo. . .	88
4.34. Transcodificador en cascada.	89
4.35. Situación de congestión en la red por falta de ancho de banda durante el acceso al servicio IPTV.	92
4.36. Solución de la congestión con la agrupación de grupos multicast. .	92
5.1. Estructura de la red neuronal basada en unidades producto	98
5.2. Estructura de la máquina de soporte vectorial para regresión. . . .	102
5.3. Función pérdida para la dispersión el los vectores soporte en un ejemplo de regresión lineal.	104
6.1. Algoritmo de gestión basado en la QoE	113
6.2. Selección del códec en base a las características del sistema.	115
6.3. Selección del códec en base a las características del sistema simplificado.	116
6.4. Promedio de la VQM con un retardo en la red de 20ms.	117
6.5. Promedio de la VQM con un retardo en la red de 200ms.	117
6.6. Promedio de la VQM con diferentes valores de pérdidas de paquetes en la red.	118
6.7. Promedio de la VQM con diferentes anchos de banda de red. . . .	118

6.8. Protocolo de gestión para un servicio de distribución de red de IPTV para redes inalámbricas	122
6.9. Proceso de gestión del cliente basado en QoE y RSSI para servicio de IPTV en redes inalámbricas	124
6.10. Algoritmo del controlador de vídeo para servicio de IPTV en redes móviles.	127
6.11. Algoritmo del cliente IPTV en redes móviles.	128
6.12. Protocolo de red en una transmisión móvil.	129
6.13. Retardo de H264 en UMTS.	130
6.14. Retardo de MP4 en UMTS.	130
6.15. Retardo de XVID en UMTS.	131
6.16. Jitter de H264 en UMTS.	131
6.17. Jitter de MP4 en UMTS.	131
6.18. Jitter de XVID en UMTS.	131
6.19. Pérdida de paquetes de H264 en UMTS.	132
6.20. Pérdida de paquetes de MP4 en UMTS.	132
6.21. Pérdida de paquetes de XVID en UMTS.	133
6.22. Retardo de H264 en WIFI.	133
6.23. Retardo de MP4 en WIFI.	133
6.24. Retardo de XVID en WIFI.	133
6.25. Jitter H264 en WIFI.	134
6.26. Jitter de MP4 en WIFI.	134
6.27. Jitter de XVID en WIFI.	135
6.28. Pérdida de paquetes de H264 en WIFI.	135
6.29. Pérdida de paquetes MP4 en WIFI.	135
6.30. Pérdida de paquetes de XVID en WIFI.	135
6.31. Algoritmo de gestión para la mejora en la entrega de vídeo 3D/ST .	138

6.32. Comparación de la VQM de diferentes flujos de vídeo con distintos formatos de codificación.	141
6.33. Jitter en la red del cliente durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.	142
6.34. Jitter en la red del cliente durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.	142
6.35. Retardo en la red del cliente durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.	143
6.36. Retardo en la red del cliente durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.	143
6.37. Paquetes perdidos en la red del cliente durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.	144
6.38. Paquetes perdidos en la red del cliente durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.	144
6.39. Ancho de banda del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D + Z y vídeo estéreo.	145
6.40. Ancho de banda del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.	145
6.41. PSNR del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.	146
6.42. PSNR del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.	146
6.43. MIV del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D + Z y vídeo estéreo.	147
6.44. MIV del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.	147
6.45. Proceso de entrenamiento para la RNRUP.	149
6.46. Proceso de test para la RNRUP.	149
6.47. Proceso de entrenamiento para la MSVR con un Kernel Polinomial.	150
6.48. Proceso de test para la MSVR con un Kernel Polinomial.	150
6.49. Proceso de entrenamiento para la MSVR con un Kernel RBF.	150

6.50. Proceso de test para la MSVR con un Kernel RBF.	150
6.51. MSE de cada una de las iteraciones realizadas con validación cruzada para RNRUP.	151
6.52. MSE de cada una de las iteraciones realizadas con validación cruzada para MSVR con kernel RBF.	151
6.53. MSE de cada una de las iteraciones realizadas con validación cruzada para MSVR con kernel Polinomial	151
6.54. Resultados de los diferentes test realizados para la evaluación del sistema de gestión en un caso práctico.	153
7.1. Diagrama de clases uml.	159
7.2. Pantalla de inicio del programa de gestión donde se introducen los datos de administración.	160
7.3. Pantalla de gestión del software	163
7.4. Algoritmo de gestión utilizado por el software	170
7.5. Datos necesarios para la conexión con el dispositivo a gestionar. . .	171
7.6. Proceso de comprobación de conexión con el dispositivo a gestionar.	172
7.7. Proceso de comprobación de conexión con el dispositivo a gestionar.	173
7.8. Topología de red tipo del banco de pruebas.	175
7.9. Ancho de banda utilizado y pérdidas de paquetes para el canal del cliente 2.	176
7.10. Estimación de la QoE para el canal del cliente 2.	177

Índice de tablas

3.1. Comparación entre calidad subjetiva y VQM	25
3.2. Posible conversión de PSNR a MOS	31
4.1. Handover entre diferentes tecnologías.	65
4.2. Tabla de códecs de vídeo Familia MPEG.	67
4.3. Tabla de códecs de vídeo serie ITU H-XXX.	69
4.4. Tabla comparativa del porcentaje de compresión de diferentes có- decs respecto a MPEG-2	72
4.5. Características de vídeo de diferentes dispositivos móviles.	74
4.6. Tabla de códecs de vídeo 3D/ST	76
4.7. Comparación códecs estereoscópicos	79
4.8. Comparación de la capacidad de compresión en tasa de bits (Kbps) para distintos valores de QS.	80
4.9. Comparativa del proceso de transcodificación	91
6.1. Simulaciones para la evaluación de los distintos formatos de codifi- cación y el MOS.	140
6.2. Comparativa entre RNRUP y MSVR a nivel de MSE y STD tanto para entrenamiento como para test	152
6.3. Descripción de las condiciones de red por cada test realizado. . . .	152

Capítulo 1

Introducción

En este capítulo, en la sección 1.1, haremos una introducción al trabajo de investigación y desarrollo llevado a cabo comentando los principales aspectos del mismo. En la sección 1.2 enumeramos los objetivos así como las motivaciones que nos planteamos a la hora de realizar esta tesis doctoral. En la sección 1.3 comentaremos los precedentes que ha habido a este trabajo y por último en la sección 1.4 explicaremos cuál es la estructura en la que se ha distribuido la memoria.

1.1 Introducción

Actualmente existe un aumento de ISP que ofrecen servicios a los hogares tales como datos, Voice on IP (VoIP), IPTV y 3D-IPTV. Esto es debido principalmente a la mejora en las infraestructuras de red que permiten un mejor acceso a Internet de banda ancha. Por tanto, una de las principales cuestiones a tener en cuenta por el proveedor de servicios es garantizar la QoE percibida por el usuario final. Por este motivo, la medida de QoE está tomando gran importancia, tanto en el mundo científico como en el empresarial, ya que es la forma que tiene el proveedor de valorar la satisfacción por parte del usuario del servicio entregado.

Algunas de las empresas que están en esta línea de investigación son: Movistar Imagenio, Orange TV, Jazztelia TV, etc... Debido a la experiencia que nuestro grupo de investigación tiene en este campo, la empresa Orange decidió ponerse en contacto con nuestro grupo de investigación a raíz de nuestras publicaciones sobre IPTV con el fin de iniciar un proyecto de investigación y desarrollo. La empresa le planteó un problema de gestión en su servicio de IPTV basado en la necesidad de garantizar la QoE del usuario final y que derivó en este trabajo de investigación. El trabajo se ha dividido en una primera fase de investigación y una segunda

de desarrollo. En la fase de investigación se ha llevado a cabo el estudio de los parámetros que influyen en la QoE del usuario final en distribuciones de IPTV, tanto en entornos de red inalámbricos, como cableados, como móviles. Uno de los aspectos importantes en la investigación ha sido estudiar cómo los parámetros de QoS así como los formatos de codificación en una transmisión de flujo IPTV influyen en la QoE, entendiendo por parámetros de QoS principalmente el jitter, retardo, paquetes perdidos y ancho de banda (Bandwidth (BW)). Tras el periodo de investigación, se pasó a la fase de desarrollo de la herramienta. Esta herramienta debía cumplir con las expectativas de gestionar el servicio de IPTV basándose en la QoE. Tras su implementación se llevó a cabo una demostración en la propia empresa de FTE en la que se valoró muy positivamente la utilidad y la capacidad de dicha herramienta. De esta forma finalizó esta fase del proyecto lo que dio lugar al inicio de una nueva fase basada en el desarrollo de un sistema de gestión basado en técnicas de inteligencia artificial.

En nuestro afán por aumentar nuestra experiencia en este campo se decidió llevar a cabo la siguiente fase de investigación. Debido a que el sistema desarrollado era un sistema heurístico se pensó en mejorarlo añadiendo sistemas o métodos de inteligencia artificial. Con ello añadimos autonomía de decisión al sistema así como la capacidad de aprendizaje en base a la experiencia de nuevos sucesos.

La integración de un método inductivo de predicción al sistema de gestión nos supuso desarrollar en primer lugar un Corpus con el que poder llevar a cabo la parte de aprendizaje y test. Las muestras que conforman el corpus se extrajeron de los diferentes estudios realizados. Los estudios se fundamentaron principalmente en el análisis de las transmisiones de flujos de vídeo en condiciones de degradación de la QoE por factores como: valores inadecuados de los parámetros de red en diferentes entornos (cableado, inalámbrico y móvil), la codificación (incluido 3D/ST), la cuantificación, la transcodificación, las características de los dispositivos de recepción y factores de las propias condiciones de entorno como pueden ser el proceso de handover. A partir de los resultados obtenidos, se crearon vectores de características por cada prueba realizada con valores correspondientes a parámetros de QoS, QoE y formatos de codificación. Cada vector estaba asociado a una etiqueta de clase correspondiente al valor estimado de la QoE. Con esta información se creó el corpus de trabajo.

Conocer la QoE de un usuario en tiempo real sería una información muy valiosa para un sistema de gestión. No obstante una medida como el MOS (Mean Opinion Score (MOS)) no puede ser obtenida en tiempo real por ser imprescindible la intervención humana. Nuestra propuesta consiste en utilizar el corpus de entrenamiento descrito para estimar la QoE a partir del estado actual de la red y configuración del sistema. Es decir necesitamos obtener una función que tome como entrada los parámetros de QoS y formato de codificación utilizado y tenga como salida una estimación de la QoE. Este tipo de problemas se conocen en el campo de la inteligencia artificial como regresión. Los sistemas de regresión que

pensamos que se ajustaban mejor al problema fueron el de Redes Neuronales de Regresión (RNR) y el de Máquinas de Soporte Vectorial de Regresión (MSVR). Ajustamos ambas técnicas al problema. Por los resultados de estimación obtenidos seleccionamos finalmente RNR. Por último, el método de RNR fue integrado al sistema de gestión inteligente. De esta forma, dada una situación de degradación de la QoE durante una transmisión de flujo de vídeo, se utiliza RNR para determinar con qué formato de codificación se obtiene un mejor valor de QoE y en base a este resultado transcodificar el flujo de vídeo transmitido.

1.2 Objetivos y motivación

El principal objetivo de este trabajo es diseñar y desarrollar un sistema de gestión basado en técnicas de inteligencia artificial para la selección de la codificación óptima en la transmisión de vídeo sobre IP, en diferentes entornos de red, que garantice la QoE del usuario final. Para llevar a cabo este objetivo principal nos planteamos los siguientes tres objetivos operativos: creación de un corpus de trabajo, ajuste de un método inductivo de predicción de la QoE y por último la integración del método de predicción en el sistema de gestión. Como último objetivo ha sido demostrar de forma práctica, a partir de los resultados obtenidos con el sistema desarrollado, los beneficios que dicho sistema aporta.

El estudio realizado se basa principalmente en el desarrollo de un método inductivo de predicción. El resultado ha sido un sistema de gestión inteligente que garantiza la QoE en la transmisión de flujos de vídeo y TV. Para el desarrollo del sistema de gestión inteligente se han utilizado técnicas de inteligencia artificial de regresión con el que hemos creado modelos adaptados al problema planteado. Con el fin de conseguir este objetivo principal se llevaron a cabo los siguientes objetivos secundarios.

El primer objetivo operativo que nos planteamos fue crear un Corpus con el que poder trabajar en el proceso de desarrollo del método inductivo de predicción. Para ello analizamos todos aquellos parámetros que pensamos influyen en la degradación de la QoE, tanto a nivel de red como de usuario, durante una transmisión de flujo de vídeo sobre IP. A nivel de red analizamos transmisiones de flujo de vídeo estándar, HD y 3D/ST sobre entornos de red inalámbricos (Wireless Fidelity (WIFI) y Worldwide Interoperability for Microwave Access (WIMAX)), cableados (Ethernet) y de telefonía móvil, con el fin de comprobar cómo se ven afectados los parámetros de red durante una transmisión de flujo de vídeo al variar las técnicas de codificación y el entorno de red. A nivel de usuario analizamos como se veía afectada la calidad de vídeo recibida en cada uno de estos casos. A partir de toda esta información extrajimos por cada una de las pruebas realizadas un vector de características con valores de: jitter, delay, paquetes perdidos, ancho de banda. Además de estos valores, también se añadió al vector de características la técnica

de codificación utilizada en cada prueba. Para etiquetar cada muestra se utilizaron métodos objetivos y subjetivos de evaluación. Para ello se compara el vídeo origen con el recibido después de haber sido transmitido y se obtiene una estimación de la QoE. De esta forma configuramos un Corpus compuesto por muestras a partir de vectores de características con valores de parámetros de red y de la técnica de codificación utilizada etiquetados en base a estimaciones de la QoE.

El siguiente objetivo operativo fue encontrar el método de regresión que mejor se adaptara al problema. Con el método inductivo de predicción buscamos estimar la QoE dadas unas condiciones de transmisión. Para su ajuste llevamos a cabo un proceso de aprendizaje supervisado y test utilizando el Corpus desarrollado. Una vez seleccionado el método inductivo de predicción se pasó al siguiente objetivo.

Otro objetivo operativo que se llevó a cabo consistió en la integración de dicho método inductivo de predicción al sistema de gestión. Cuando el sistema de gestión detecta una situación de degradación de la QoE intenta solucionarlo de dos formas: primero mediante el reajuste de los parámetros de red y segundo mediante la transcodificación. Si se lleva a cabo el proceso de transcodificación inteligente, se busca de entre todos los formatos de codificación estudiados con cual se obtendría una mejor estimación de la QoE. Con el formato resultante se realizaría la transcodificación sobre el flujo de vídeo que se está transmitiendo.

Una de las motivaciones de este estudio partió de la realización de un proyecto en colaboración con la empresa Orange. En este proyecto se llevó a cabo el desarrollo de un sistema de gestión basado en métodos heurísticos de decisión. El sistema fue integrado en un entorno real. En base a este proyecto se pensó en mejorar dicho sistema cambiando el método heurístico por un método de decisión basado en inteligencia artificial. La finalidad es integrarlo también en un entorno real con el fin de mejorar la eficiencia en la QoE ofrecida por el proveedor de servicios.

Otra de las motivaciones que nos ha llevado al desarrollo de un método de estimación de la QoE de forma objetiva es que pensamos que sería de grandísimo interés para el avance científico en este campo. El motivo es que dependemos continuamente de la inversión de un importante despliegue de recursos a la hora de valorar un sistema basado en la QoE. Se necesitan expertos en el tema o encuestas a usuarios que valoren la calidad del vídeo recibido. Se necesita un número considerable de opiniones para llegar a una aproximación fiable, por lo que se tienen que realizar varios tests para una misma prueba. Esto se complica cuando las valoraciones se hacen sobre vídeos 3D/ST, ya que se necesita una especialización todavía mayor. A parte de este planteamiento lanzamos la siguiente cuestión: ¿No es verdad que la valoración subjetiva no deja de ser subjetiva? Es decir, lo que para un cliente la calidad es una cosa para otro no lo es. Es cierto que hay una línea de aproximación entre todos pero no deja de haber un error. Pensamos que si planteamos un sistema que se aproxime con gran acierto a la valoración que podría hacer un

experto, siendo conscientes que habría siempre un margen de error, sería un gran avance.

1.3 Precedentes

En este trabajo de investigación partimos de la base de que los parámetros de red influyen en la QoE. Artículos como (Kim y col. 2008a) y (Kim y Choi 2010) nos ayudaron a reafirmarnos en nuestra idea. Los autores en (Kim y col. 2008a) proponen un método objetivo para medir de forma aproximada la QoE mediante parámetros de QoS. Parten de la idea de que la QoS está estrechamente relacionada con la QoE. A partir de ahí extraen un modelo de correlación entre la QoE y la QoS con el que se demuestra la influencia entre ellas. Muy parecido es el planteamiento realizado en (Kim y Choi 2010), en el que proponen asociar la QoE con parámetros de calidad de servicio como el retardo, pérdida de paquetes, jitter, etc...

Uno de nuestros objetivos en esta tesis doctoral ha sido estimar la QoE de forma objetiva ya que por un lado los recursos necesarios para las evaluaciones subjetivas de la QoE son costosos y por otro pensamos que sería de gran ayuda en el avance científico en este campo. En relación a este planteamiento han trabajado también en (Agboma y Liotta 2008). En este artículo los autores emplean una técnica basada en un modelado estadístico en el que se correlaciona los parámetros de QoS con las estimaciones de las percepciones de la QoE. Este método identifica el grado de influencia de cada uno de los parámetros de QoS en la percepción del usuario. Aunque incitan a utilizar este planteamiento en algún sistema de gestión no llegan a realizarlo, dejando un poco en el aire su aplicabilidad.

Durante la entrega de servicios IPTV uno de los principales problemas que se plantean son las transmisiones en entornos de redes inalámbricas, de redes móviles y actualmente las transmisiones 3D/ST. En estos casos es más probable que se de una degradación de la QoE y no siempre es posible solucionarlo ajustando los parámetros de red. Por este motivo, en el desarrollo de nuestro sistema de gestión planteamos la transcodificación como una posible solución a este problema. Hay trabajos que respaldan esta propuesta como en (Liu y col. 2012), en el que se propone un enfoque orientado a la transcodificación en base a la QoE para mejorar la calidad del servicio de vídeo móvil en 3D. Un libro muy interesante en el que se analiza el proceso de transcodificación en IPTV y que utilizamos de referencia para valorar la factibilidad de nuestro sistema es (O'Driscoll 2008). Como apoyo a este último libro y para profundizar más en la transcodificación en entornos de red inalámbricos utilizamos (Ahmad 2008).

Unos de los primeros estudios que se realizaron sobre la gestión basada en QoE y QoS fueron sobre redes Universal Mobile Telecommunications System (UMTS) para sistemas de telefonía. Trabajos basados en este tema, como el libro (Soldani,

Li y Cuny 2007) y como el artículo (Yaici y col. 2005), nos sirvieron para recabar ideas para llevar a cabo el desarrollo de nuestro sistema de gestión para IPTV. Nuestras primeras propuestas, como en (Watanabe y col. 2008), se basaron en un sistema de gestión para servicios de IPTV basados en la QoE en un contexto de redes de nueva generación. Esta primera propuesta era básicamente analítica, en las que a partir de una serie de estudios realizados obtuvimos los valores de los parámetros de red que garantizaban de forma aproximada la QoE del usuario. Este sistema de gestión ha sido mejorado utilizando técnicas de inteligencia artificial para la predicción de la QoE. A partir del desarrollo de un método inductivo de predicción somos capaces de predecir la QoE del usuario final. Con este resultado el sistema de gestión decide qué proceso de gestión es el adecuado. Recientemente hay trabajos en esta línea como el de (Agboma y Liotta 2008). En él los autores proponen una metodología de gestión con el fin de proporcionar diferentes niveles de calidad de servicio. Utilizan modelos estadísticos para establecer umbrales en los que la percepción del usuario de la calidad de servicio se convierte en inaceptable. Además analizaron también el grado de influencia de cada uno de los parámetros de QoS en la percepción del usuario. Nuestro enfoque se basa en métodos de inteligencia artificial, lo que le da la cualidad de adaptativo con posibilidad de aprender en base a la experiencia.

Como acabamos de comentar uno de los aspectos importantes en nuestro sistema de gestión es el método inductivo de predicción. Dos trabajos que influyeron en la selección de esta metodología fueron (Frank e Incera 2006) y (Mohamed y Rubino 2002). En (Frank e Incera 2006) proponen una herramienta basada en redes neuronales para la evaluación de la calidad de flujos de vídeo en redes IP. Es capaz de predecir la percepción humana utilizando parámetros objetivos. En (Mohamed y Rubino 2002) plantean un enfoque similar. Utilizan redes neuronales aleatorias. La ventaja de esta metodología es que es capaz de cuantificar la calidad de vídeo automáticamente. Un libro que nos ha sido de mucha ayuda para realizar cualquier consulta relacionada con el análisis y desarrollo del método de predicción ha sido (Bishop 1995).

1.4 Estructura de la memoria

La estructura de la memoria viene definida de la siguiente forma:

En el capítulo 2 hemos presentado el estado del arte sobre los temas relacionados con nuestro trabajo en base a: la arquitectura de red para IPTV, métodos de codificación y transcodificación, algoritmos de gestión y sistemas de gestión.

En el capítulo 3 hemos definido algunos conceptos generales para la mejor comprensión del trabajo de investigación. Hemos definido términos como QoE, QoS, métodos objetivos y subjetivos de evaluación de la calidad de vídeo, MOS, etc...

También hemos descrito aquellos parámetros que consideramos afectan a la QoE y QoS así como aspectos globales que se han utilizado en la memoria como: la infraestructura general de red para IPTV, el algoritmo de gestión global y el banco de pruebas utilizado en cada uno de los estudios.

En el capítulo 4 analizamos todas aquellas medidas que se han llevado a cabo en el desarrollo del sistema de gestión. En dicho capítulo hemos estudiado aspectos tan determinantes en el desarrollo del sistema de gestión como son la transcodificación y la cuantificación. Hemos analizado también las principales características de los estándares de compresión de la familia Moving Picture Experts Group (MPEG) y la serie International Telecommunication Union (ITU) H-XXX., de los formatos de codificación para contenedores multimedia, para dispositivos móviles y para 3D/ST. Otro de los aspectos que necesitábamos analizar para el desarrollo del sistema de gestión es el proceso de handover y que ha sido incluido en este capítulo.

En el capítulo 5 hemos realizado un análisis teórico de los diferentes métodos de predicción de la QoE utilizados para el desarrollo del sistema de gestión. También hemos analizado de forma teórica el método de selección del formato de codificación óptima en base a los anteriores métodos de predicción como método fundamental del sistema de gestión.

En el capítulo 6 hemos analizado cada uno de los procesos que realiza el sistema de gestión para la entrega de un servicio de IPTV. Estos procesos son: activación del sistema de gestión, gestión por degradación de la QoE (modificación de los parámetros de red, transcodificación heurística o transcodificación inteligente), gestión para entornos cableados, inalámbricos o móviles y gestión en transmisiones con formatos de flujo de vídeo estándar, High Definition (HD) o 3D/ST.

En el capítulo 7 hemos descrito el software de gestión basado en el método inductivo de estimación. Hemos explicado su funcionamiento y hemos demostrado su aplicabilidad con un ejemplo.

Por último, en el capítulo 8 se hemos discutido cada una las conclusiones a las que se ha llegado tras finalizar el trabajo de investigación además de indicar las futuras líneas de trabajo.

Capítulo 2

Estado del arte

En este capítulo se describe el estado del arte de aquellos aspectos que caracterizan nuestro trabajo salvo en la sección 2.1, en la que se ha realizado una introducción tratando aspectos más generales de la tesis. En la sección 2.2 se ha buscado en la literatura planteamientos relacionados con la arquitectura de red IPTV. En la sección 2.3 se ha buscado información con respecto al proceso transcodificación aplicado en IPTV. En la sección 2.4 se ha investigado sobre algoritmos de gestión para IPTV y en la sección 2.5 sistemas de gestión para IPTV y se han comparado con nuestro trabajo.

2.1 Introducción

El aumento y la mejora en las tecnologías así como en las prestaciones en ancho de banda, codificación y protocolos han hecho que mejore la calidad en servicios como IPTV. Como consecuencia ha habido un aumento considerable de la demanda de este tipo de servicios. Según el grupo de expertos de la Unión Internacional de Telecomunicaciones (UIT) en IPTV, (*IPTV Focus Group*), afirman que el servicio de IPTV debe poseer una adecuada QoS, seguridad, interactividad y fiabilidad. Es por eso que el servicio de IPTV debe tener una correcta QoS y una adecuada QoE para satisfacer las necesidades de los usuarios finales. Así pues, se requiere tecnologías de transmisión, como la multicast, junto con algoritmos de gestión (Jimenez y col. 2015) que solucionen problemas de congestión, retardo, jitter, paquetes perdidos, etc... y así garantizar la QoE. Uno de los principales problemas de la oferta multicast de televisión son los cuellos de botella. Esto, se produce por ejemplo cuando hay una petición masiva de un mismo canal de IPTV. Para solucionarlo es necesario multiplexar la transmisión multicast y utilizar el sistema de codificación adecuado (Ng, Liew y Lin 2005). Una desafortunada codificación del

flujo de vídeo podría afectar claramente a la QoE así como a la compatibilidad con los navegadores, con los frames por segundo de algunos países, con los sistemas operativos, etc... tal como se ha analizado en (Lopez-Herreros y col. 2015). En (Imran, Mellia y Meo 2007) proponen la combinación de Constant Bitrate (CBR) y Variable Bitrate (VBR) así como el número ideal de Intra-Imágenes dentro de la secuencia Group of Pictures (GOP) para confeccionar un flujo de tramas más eficaz. Todos estos son factores que deberemos tener en cuenta en el desarrollo de nuestro sistema de gestión.

Un sistema eficiente de entrega de IPTV o VoD debe de ser capaz de garantizar la QoE. Para ello se debe de tener en cuenta parámetros con respecto a la transmisión de flujo de vídeo como son: la pérdida de paquetes, la tasa de bit, etc. En el reciente trabajo (Juluri, Tamarapalli y Medhi 2016) han estudiado como a partir de las cabeceras de los paquetes (TCP, RTP, UDP, IP etc.) se puede obtener este tipo de información durante la transmisión. Además han hecho una recopilación de actuales técnicas de entrega de vídeo VoD, métricas de QoE, métodos de medición de la QoE y modelos predictivos de la QoE.

El uso de la TV digital, a través de transmisiones de vídeo digital (Digital Video Broadcasting (DVB)) (*DVB worldwide*), dio lugar a otros estándares de difusión de TV como: TV móvil (Carlsson y Walden 2007) y transmisiones de vídeo digital portátiles (Digital Video Broadcasting Handheld (DVB-H)). Actualmente, la entrega de televisión móvil para teléfonos, tablets y dispositivos móviles está teniendo gran éxito. *Yamgo TV* es uno de los proveedores de este tipo de servicios. Para poder ser integrados se necesita de los requisitos recomendados en la (ITU-T/FG/IPTV-DOC-0147 2007). También son necesarias técnicas relacionadas con la monitorización del tráfico en tiempo real en este tipo de redes con el fin de garantizar tanto la QoS como la QoE (Gupta y Dureja 2015). A mediados del 2012 todavía no se había publicado ningún sistema que determinara los parámetros de QoS en una red móvil de IPTV necesarios para garantizar la QoE del usuario final durante transmisiones entre dispositivos móviles y redes multimedia. Este tema ha sido tratado con gran profundidad por nuestro grupo de investigación en (Lloret y col. 2011b).

La necesidad de itinerancia entre diferentes tecnologías inalámbricas es relativamente reciente. Con el fin de solucionar este problema se desarrolló el estándar (IEEE/STD/802.21 2008) que fue publicado en Noviembre del 2008. Su objetivo es proporcionar mecanismos que optimicen el paso entre redes IEEE 802 heterogéneas sin mermar la QoE del usuario. Uno de estos mecanismos consiste en añadir inteligencia a nivel de la capa de enlace. En (Dai y col. 2008) los autores presentan un eficiente mecanismo de traspaso-vertical (*vertical-handover*) entre redes WiMAX y WIFI sin que esto implique ningún cambio en la red. Proponen un sistema basado en 2 avisos o disparadores (*triggers*): uno indica si la conexión se va a perder y el otro si el rendimiento, dado por la velocidad de datos y de la carga de la red, no es el idóneo. Cabe decir que esta propuesta no garantiza la

QoS. Sin embargo es de mucho interés para nosotros dado que si existe un usuario que requiere mayor ancho de banda cabrá la posibilidad de cambiarlo a otra red que cumpla con los requisitos, aunque sea de otra tecnología diferente. Este planteamiento ha sido adaptado a nuestro sistema de gestión.

En los últimos años, las tecnologías de acceso inalámbricas de banda ancha han evolucionado espectacularmente. Su alcance va desde los estándares IEEE 802 a los sistemas 3GPP-LTE y 4G (Ohmori, Yamao y Nakajima 2000). Todos ellos han estado compitiendo por llegar a ser la tecnología inalámbrica más importante mejorando cada vez más sus características y rendimiento. Los sistemas 4G, basados completamente en IP, asumen la combinación de varias tecnologías como la mejor opción para proporcionar acceso inalámbrico de banda ancha para redes móviles. Avances en los que se sugieren la interoperabilidad mundial para acceso por microondas (WiMAX) y Long Term Evolution (LTE) son la base tecnológica para la implementación de sistemas 4G (Kuran y Tugcu 2007). Una parte de este trabajo de investigación ha consistido en comprobar el rendimiento de transmisiones de flujo de vídeo a través de entornos de telefonía. Servicios multimedia para este tipo de entornos destaca la tecnología Multimedia Broadcast Multicast Service (MBMS) (Holma, Kristensson y Kaikkonen 2010). MBMS se centra en la difusión multimedia en redes 3G, como podría ser la distribución de televisión a través de un canal compartido por todos los usuarios que están viendo un programa en la misma zona. MBMS complementa el Speed Downlink Packet Access High (HSDPA) (Chaudhry y Khan 2009) mediante el soporte de la carga en zonas densas (con muchos usuarios) y asegura el uso eficiente del ancho de banda de la red.

Uno de las recientes líneas de investigación en IPTV es el estudio de la entrega de servicios 3D/ST. Al ser un tema reciente se plantean dudas como: ¿cuál es mejor sistema para el transporte sobre redes IP en tiempo real para la transmisión de este tipo de flujos de vídeo? (Kim y col. 2008c) (Petrovic, Farin y With 2008), ¿Qué tipo de dispositivos y tecnologías de reproducción 3D/ST a nivel de usuario responden mejor a este tipo de entornos? (Saygili, Gürler y Tekalp 2009) (Hewage y col. 2007), ¿Cómo se debería de evaluar la calidad de este tipo de vídeos? (El-Yamany y col. 2010) , ¿Qué arquitectura de red se podría ajustar a este contexto?(Matusik y Pfister 2004), ¿Qué técnicas de codificación se deberían de utilizar con el fin de reducir el ancho de banda necesario?, etc... Todo ello son preguntas que nos hemos planteado para el desarrollo de este trabajo con el fin de garantizar la QoE del usuario final.

2.2 Arquitectura de red para IPTV

Triple play (Hellberg, Greene y Boyes 2007) y Quad play (inCode Advisors 2006) son servicios integrados utilizados sobre el protocolo IP. Por un lado, triple play integra tres servicios: voz, datos de alta velocidad y televisión. Por otro lado, cuádruple play es el triple play, más la movilidad de los usuarios. Para soportar estos servicios correctamente, las redes están evolucionando según las arquitecturas de Next Generation Networking (NGN) (Knightson, Morita y Towle 2005). La idea general del concepto de NGN es que la red transporta toda la información y los servicios, tales como voz, datos y vídeo, encapsulados en paquetes con el Internet Protocol (IP). Además, el transporte debe ser completamente independiente de la infraestructura de red que se utilice. NGN toma en cuenta la QoS (Schollmeier y Winkler 2004) para proporcionar servicios multimedia con una calidad aceptable a través de redes no orientadas a conexión. La mayoría de las tecnologías inalámbricas de los administradores de red están incorporando un IP Multimedia Subsystem (IMS) (Saleem y Mohan 2007) adaptándose así a las NGN. Esto simplifica la interfaz entre las aplicaciones de IPTV y otros servicios IP tales como la conexión de datos de alta velocidad y VoIP. El IMS es una parte emergente de la arquitectura de red que permite a los operadores acelerar y simplificar el despliegue de servicios basados en IP. La arquitectura de red que planteamos es perfectamente adaptable a las NGN ya que a nivel de transporte la infraestructura se puede basar en el ruteo de paquetes a través de direcciones IP incorporando IMS y a nivel de transporte planteamos técnicas de gestión para la solución de problemas basadas en la prioridad de tráfico que perfectamente podrían ser implementadas en IP/MPLS (Multiprotocol Label Switching (MPLS)). El sistema de gestión no se vería afectado.

En el diseño de la arquitectura de red hemos tenido en cuenta que es necesario ofrecer características como escalabilidad (García y col. 2009), QoS (Canovas y col. 2012), alto soporte al tráfico Transmission Control Protocol (TCP) (Atenas y col. 2010), etc... ya que tal como han demostrado los autores en (Agrawal y col. 2007), el futuro del tráfico de Internet estará dominado por flujos de vídeo y aplicaciones multimedia bajo demanda como IPTV, vídeo estereoscópico/HD, juegos, realidad virtual, etc...

Actualmente ya existen múltiples sistemas de IPTV, tanto privados como propietarios, que son utilizados por los distribuidores de IPTV. Concretamente, en el caso de estudio realizado por (Cisco 2007), han desarrollado una arquitectura para transportar vídeo y reducir el tiempo de envío, de manera eficiente. Actualmente se está utilizando Cisco IPTV multicast en videoconferencias (e-learnings) y Video on Demand (VoD). Para solucionar el cuello de botella se multiplexan como máximo 50 usuarios simultáneamente. Si se supera ese número se cambia la codificación de las tramas tanto de audio como vídeo. La infraestructura que plantea Cisco para este tipo de servicios requiere de servidores de flujo de vídeo,

redes troncales para multicast a 100, 300, 900 Kbps, una red Cisco Application and Content Networking System (ACNS) así como un servicio al cliente a través de Local Area Network (LAN), WIFI, 3G o WiMax y Virtual Private Network (VPN) para acceso remoto de los trabajadores. En nuestros estudios, también planteamos técnicas de multiplexación en transmisiones multicast para la solución del cuello de botella, (Atenas y col. 2010) y (Lloret y col. 2011b), pero el verdadero potencial de nuestra arquitectura está en la cabecera de red. En ella planteamos una granja de servidores y un controlador de vídeo desde donde se gestiona todo el tráfico de red y desde donde se soluciona cualquier problema.

Las transmisiones 3D/ST para IPTV son un hecho. Los autores, en (Li, Magnor y Seidel 2003), muestran la implementación de un sistema de 3D Television (3DTV) sobre el protocolo Transmission Control Protocol/Internet Protocol (TCP/IP) capaz de: la adquisición en tiempo real, transmisión y visualización de imágenes 3D de escenas dinámicas. También presentan la calibración de la imagen y el proceso de alineación necesario para lograr una buena calidad de imagen para 3DTV. Para demostrar su factibilidad llevaron a cabo un estudio de la QoE.

La transmisión de flujos de vídeo estereoscópico sobre IP en las actuales arquitecturas de red presentan un nuevo reto ya que tienen que ser transmitidos a todos los receptores que lo soliciten y el ancho de banda necesario es limitado. Esto plantea un problema a la hora de garantizar una apropiada QoS en la capa de transporte tal como se justifica en (A. M. Tekalp y Civanlar 2007). Los autores de este trabajo ofrecen dos posibles soluciones a este problema: por un lado, proponen mejorar la escalabilidad a nivel de red (por ejemplo, implementando multicast nativo), donde los elementos de la red, como los routers, replican todos los paquetes enviados a los usuarios finales. Por otro lado, proponen varios servidores de televisión utilizando la tecnología Peer-to-Peer (P2P) a través de Internet. G. Petrovic presenta en (Petrovic, Farin y With 2008) una posible implementación de un sistema de flujo de vídeo estereoscópico sobre una red IP. La arquitectura del sistema de flujo se basa en varias capas, donde cada capa transmite información a partir de una sola señal de vídeo codificada. Este sistema podría ser implementado con el hardware y el software existente. En nuestra arquitectura nos decantamos por mejorar la escalabilidad a nivel de red implementando el multicast nativo ya que con la tecnología P2P no es posible realizar una administración de red de forma centralizada lo que dificulta determinar quién controla los recursos de la red y éstos son factores indispensables en nuestro sistema de gestión, ya que el controlador del sistema de gestión debe ser capaz de monitorizar de forma centralizada los recursos de red para determinar de forma inteligente la codificación óptima para este tipo de transmisiones.

Otro aspecto a tener en cuenta es la seguridad en la transmisión del contenido multimedia. Por ejemplo en (Nishimoto, Imaizumi y Mita 2009) proponen un esquema de gestión de derechos digitales integrados para la difusión móvil que permita una transmisión segura. En (Meyer 2006) proponen algunas soluciones para llevar a

cabo la itinerancia entre proveedores y entre sistema de handover. En (Shi, Shen y Mark 2004) proponen una arquitectura segura para llevar a cabo la itinerancia tanto en redes LAN como en redes de telefonía móvil. La seguridad es un aspecto que no se ha tratado en este trabajo pero lo propondremos como futura mejora.

El paso de una red inalámbrica a otra es un aspecto a tener en cuenta en el análisis de la QoE. En (Ma y col. 2004) los autores presentan un nuevo método basado en el protocolo de transmisión para el control de flujo Stream Control Transmission Protocol (SCTP), para llevar a cabo vertical handover entre diferentes redes. Este trabajo demuestra como interconectar redes UMTS y Wireless Local Area Network (WLAN) utilizando dicho método. En nuestro trabajo centramos el estudio en el proceso de handover entre tecnologías del mismo y cuál es su impacto sobre la QoE (Canovas y col. 2012). Además incluimos en el sistema de gestión un proceso para mejorar la QoE en base a la potencia de la señal recibida Received Signal Strength Indicator (RSSI)(Lloret y col. 2013).

Hay muchas obras en la literatura en las que se presentan arquitecturas o nuevos sistemas de conexión basados principalmente en los niveles de QoS. El artículo (Zhang y Liang 2008) presenta una arquitectura de banda ancha inalámbrica que soporta QoS para IPTV. Esta arquitectura se adapta al estado de la red mediante el uso de un mecanismo de control de QoS con el fin de proporcionar un servicio de IPTV con QoS garantizado. En (Park y Choi 2007) proponen un mecanismo de red donde el control de admisión de conexión se controla de acuerdo con el ancho de banda disponible. Si el ancho de banda es suficiente para asignar un nuevo flujo, se acepta la conexión con garantías de QoS. El problema es que estas políticas no se pueden aplicar cuando hay varios tipos de tráfico que tienen diferentes niveles de QoS. En (Lee y col. 2008), los autores proponen un servicio de IPTV con garantía de calidad de servicio similar al trabajo anterior. Proponen esta prestación de servicios mediante el uso de un tráfico diferenciado. Los autores proponen un esquema de ingeniería de tráfico que prioriza el tráfico IPTV. Esta priorización de tráfico se proporciona mediante la asignación de categorías de acceso diferenciadas a cada paquete de acuerdo a una clase de QoS predefinido. En nuestro trabajo el planteamiento es diferente, buscamos una correlación entre la QoE y la QoS. La QoS dependerá de la QoE y viceversa. Hay trabajos en la literatura que llevan a cabo planteamiento similar al nuestro como en (Pyykkö y col. 2006) y (Teyeb y col. 2006). Miden la QoS de forma subjetiva, es decir en base a la percepción audiovisual por parte del usuario. Nosotros vamos aún más allá de este planteamiento ya que nosotros no necesitamos de la percepción del usuario sino que es un método inductivo de predicción el que nos determinará dicha percepción.

2.3 Proceso de transcodificación en IPTV

La oferta multimedia a través de multicast ofrece una serie de mecanismos en función de la tecnología utilizada para la difusión: seleccionar diferentes canales dinámicamente y administrar el ancho de banda disponible en servicios de IPTV. Cada una de estas opciones multicast responde a un estándar y necesita de la modulación adaptativa y de la codificación correspondiente para poder llevar a cabo la emisión (Poon, Lo y Feng 2005). La transmisión de vídeo a través de la pila de protocolos TCP/IP comporta la codificación de su información en las diferentes capas de dicho modelo (Banodkar y col. 2008). La codificación ha sido uno de los aspectos que habremos tratado con mayor profundidad en nuestro trabajo de investigación ya que uno de los procesos del sistema de gestión para la solución de problemas se basa en la transcodificación.

Uno de los principales problemas en la transmisiones de vídeo 3D/ST sobre redes IP es el ancho de banda necesario. Según (Vetro y col. 2004), se requiere un ancho de banda de 5.3 Gbps para transmitir 16 secuencias de vídeo sin comprimir con una resolución de 1280x720 a 30fps. Con el fin de reducir estos requisitos en (Hewage y col. 2007) proponen una técnica de codificación asimétrica aplicada sobre el color y la profundidad de cada fotograma. El método utilizado se basa en una extensión escalable de H.264/Advanced Video Coding (H.264/AVC). Con este método se logra una alta calidad de la imagen del vídeo estéreo renderizado y un bajo bitrate comparado con otros sistemas de codificación. En (Fecker y Kaup 2004), Fecker y Kaup proponen un sistema donde el vídeo se codifica en tiempo real y utilizan el orden de la imagen transpuesta para mejorar la codificación predictiva. Otros dos sistemas de codificación para reducir el tamaño de vídeo estereoscópico se presentan en (Morton 1988) y (Hewage y col. 2007). En (Morton 1988), los autores proponen una metodología de codificación asimétrica de color y profundidad de vídeo que utiliza la arquitectura de capas de la extensión escalable de H.264/AVC. Afirman que su método logra una alta calidad de imagen con bitrates bajos. Por otra parte, se puede utilizar para escalar aplicaciones de IPTV existentes en TV estereoscópica con un incremento mínimo de los gastos y garantizando al mismo tiempo una alta calidad de la percepción de profundidad. Una forma de reducir el tamaño de los datos para la transmisión de vídeo estereoscópica es mediante el enfoque adoptado en (Hewage y col. 2007). El sistema genera una imagen de referencia de color y otra difuminada a partir de los diferentes puntos de vista. El sistema almacena las coordenadas de estos puntos en 3D. Estas coordenadas se representan como una imagen en color en 2D (24bpp) y una imagen de disparidad (16bpp). Estas imágenes se transmiten a través de la red a un render remoto. En nuestro trabajo, a nivel de codificación 3D/ST hemos trabajado principalmente con respecto al formato de codificación Stereo Video Coding (SVC) y 2D-plus-Depth (2D+Z) y al factor de escala (quantization scale (QS)) como se puede ver en (Canovas y col. 2010a) y (Lloret y col. 2013). De todas formas nosotros no buscamos un método específico de codificación para la transmisión de vídeo

3D/ST sino que buscamos aquel formato que mejor se adapte a cada una de las condiciones de red tal como mostramos en los estudios anteriormente citados.

2.4 Algoritmos de gestión

En redes móviles se requiere una eficiente distribución del contenido multimedia debido a las limitaciones en los recursos de los radio enlaces. Por ello deben de ser capaces de soportar mecanismos avanzados de control de acceso para mantener una QoS adecuada para IPTV (G. Florido 2009). En (Kim y col. 2010) analizan este tema con detalle, incluyendo transmisiones de vídeo 3D/ST. Hay un par de artículos (Lee, Kim y Song 2010) y (Lee y Song 2010) en los que utilizan algoritmos de control del canal en redes IPTV móviles. Se basan principalmente en la interoperabilidad para el acceso por microondas o en esquemas que proporcionan control de acceso. El objetivo de estos trabajos es mejorar la QoE de los abonados reduciendo el tiempo de zapping de la estación de abonado y mejorando la calidad de vídeo. Los autores, en (Rong y col. 2009), utilizan una adaptación del estándar IEEE 802.11 a nivel de la capa MAC con el fin de proporcionar una adaptación automática del contenido multimedia para cualquier dispositivo IPTV móvil. Proponen un algoritmo de programación de paquetes para priorizar el servicio IPTV móvil sobre el resto de un entorno multiservicio. Este algoritmo de control de canal permite procesar toda la información disponible en un enlace inalámbrico. En (Barry, Campbell y Veres 2001) los autores proponen servicios diferenciados en redes de paquetes inalámbricas utilizando un enfoque totalmente distribuido que soporte la diferenciación de servicios, monitoreo de radio y control de acceso. Para ello han desarrollado un algoritmo de control para IPTV móvil que permite adaptar la entrega de contenido multimedia en función de los paquetes perdidos, el jitter, el retardo, el ancho de banda disponible, el uso del servidor multimedia, el formato de la codificación de vídeo, la congestión y el proceso de handover. En (Joo y col. 2008) se presenta un algoritmo de control de canal IPTV simple pero efectivo que mantiene el tiempo de zapping en un intervalo tolerable con una alta utilización de la red. El algoritmo propuesto controla el tiempo de zapping ajustando el número de canales de radiodifusión que se encuentran cerca de los usuarios a través de redes IP y el número de tramas adicionales insertadas en cada canal, en base a la información de preferencias del canal del usuario. Finalmente, se proporcionan los resultados experimentales para demostrar el rendimiento del algoritmo propuesto.

Como podemos ver, según los trabajos del párrafo anterior, es común la utilización de algoritmos para el control de canal como medida de gestión para garantizar la QoS en redes inalámbricas y móviles. A diferencia de estas propuestas, nuestro algoritmo de gestión para este tipo de entornos se basa en garantizar la QoE y no sólo en la QoS. De esta forma, si existe una degradación de la QoE del usuario se lleva a cabo el proceso de handover. Conceptualmente, un planteamiento similar

al nuestro es el propuesto en (Joe, Kim y Hong 2007), en el que han desarrollado también un algoritmo para determinar cuándo se debe llevar a cabo el proceso de handover. El algoritmo utiliza la potencia de consumo como parámetro principal en la decisión de handover. Nosotros nos basamos también en la potencia de la señal para decidir la mejor opción de cambio de canal mediante el handover, pero es la QoE la que determina cuando.

En redes IEEE 802.11, la QoS se puede garantizar utilizando algoritmos de predicción. En (Chan y Seneviratne 1999), los autores proponen el uso de un algoritmo de predicción de movilidad del usuario adaptativo. Este algoritmo podría ser utilizado para reservar recursos y para la preconfigurar el servicio con un coste mínimo. Sin embargo, en este trabajo sólo presentan la propuesta, con algunas simulaciones y sin analizar el rendimiento real del sistema. En nuestra propuesta en lugar de predecir la movilidad predecimos la QoE del usuario. Es verdad que a diferencia de este sistema, en el nuestro necesitamos una degradación de la QoE para llevar a cabo algún tipo de acción. Sin embargo, nuestro sistema es aplicable a todos los entornos de red y no únicamente a móviles.

2.5 Sistemas de gestión

La gestión en servicios de IPTV es un aspecto fundamental ya que se debe garantizar la QoE independientemente de las condiciones de entorno, el tipo de servicio y del tipo de usuario. En (Agboma y Liotta 2008), los autores proponen una metodología de gestión con el fin de proporcionar diferentes niveles de QoS. Utilizan modelos estadísticos para establecer umbrales en los que la percepción del usuario de la calidad de servicio se convierte en inaceptable. Además analizaron también el grado de influencia de cada uno de los parámetros de QoS en la percepción del usuario. El problema de este sistema de gestión es que al ser valores heurísticos son fijos. El sistema que nosotros planteamos tiene la capacidad de aprender de la información de entorno.

Una cuestión importante con respecto a la gestión de la QoE es cómo abordar este tema para el Internet del futuro y que va en la línea de nuestro trabajo. Para ello, en (Stojanovic y col. 2015) proporcionan un resumen de las métricas y metodologías de evaluación de la QoE y posibles formas de relacionar la calidad de experiencia con parámetros de rendimiento de la red. Además, se discuten cuestiones técnicas abiertas en el logro de la QoS inter-proveedores, incluyendo mediciones de extremo a extremo a gran escala. Explican cómo la gestión centrada en el usuario asume un enfoque por capas que incluye la infraestructura de red, rendimiento de la red y la sensibilización de los usuarios. En este artículo también se hace hincapié en las oportunidades de negocio para los proveedores de servicios en el contexto de la gestión de QoE.

Una de las herramientas de nuestro sistema de gestión es el método inductivo de predicción. En (Frank e Incera 2006) proponen una herramienta basada en redes neuronales para la evaluación de la calidad de flujos de vídeo en redes IP. Es capaz de predecir la percepción humana utilizando parámetros objetivos. En (Mohamed y Rubino 2002) plantean un enfoque similar. Utilizan redes neuronales aleatorias. La ventaja de esta metodología es que es capaz automáticamente de cuantificar la calidad de vídeo. La evaluación se puede hacer en tiempo real y los resultados se aproximan a los valores que se podrían obtener en una prueba subjetiva. Nuestro enfoque es más similar al primero. Nosotros planteamos un método deductivo basado en la predicción de la QoE. Ambos trabajos se diferencian del nuestro en que estos son analíticos y nuestra propuesta además es funcional, aplicada a servicios de IPTV.

Trabajos como (Lopez y col. 2006), (Watanabe y col. 2008) y (Yaici y col. 2005) plantearon sistemas de gestión basados en procedimientos estadísticos normalizados con el fin de correlacionar la QoS y la QoE. Los planteamientos son interesantes para nuestra investigación ya que nuestro sistema se basa en la correlación entre la QoE y la QoS. Sin embargo, (Lopez y col. 2006) y (Yaici y col. 2005) no tienen un enfoque para servicios de IPTV más bien para servicios bajo demanda o para UMTS. En (Watanabe y col. 2008) presentan un enfoque global para IPTV basado en la solución de problemas de las tecnologías existentes. Por tanto no llega a ser una propuesta concreta de gestión. Un trabajo reciente relacionado con este tema es (Chervenets y col. 2016). En este artículo se ha desarrollado un modelo que correlaciona los parámetros más relevantes de QoS con factores de QoE. Es interesante que para llevar a cabo este desarrollo han presentado una recopilación de los principales métodos y recomendaciones para la correlación entre la QoS y la QoE.

Muchos de los sistemas de gestión planteados para IPTV se basan en la QoE. En (Perkis, Munkeby e Hillestad 2006) presentan un modelo para medir la QoE de los servicios multimedia móviles, teniendo en cuenta los parámetros medibles que están relacionados con aspectos tecnológicos del servicio y con parámetros no medibles que están relacionados con la percepción del servicio, sus expectativas y el comportamiento de los usuarios. El objetivo en (Perkis, Munkeby e Hillestad 2006) es producir indicadores de calidad cuantificables para la evaluación de la QoE. En (Siller 2006) los autores proponen un marco de gestión basado en la QoE, utilizando métricas de red Network Quality of Service (NQoS) y un feedback dinámico con los usuarios. Con esto, el usuario puede cambiar los requisitos de QoS en cualquier momento a través del control del feedback. Utilizando una plataforma de agentes, el feedback del usuario es mapeado en una matriz de rendimiento de referencia que refleja las necesidades del usuario.

Los autores en (Latré y col. 2009) proponen un sistema de gestión para la prestación de servicios basado en tres planos: El plano del controlador, el plano del conocimiento y el plano de acción. El plano del controlador obtiene información

a través de la monitorización de la red de acceso. Observa todos los paquetes que pasan a través de los puntos intermedios al dispositivo final y utiliza la información de estos paquetes para extraer los valores de pérdidas de paquetes, retardo y jitter. El plano de conocimiento analiza la información recibida desde el plano del controlador para detectar la degradación de QoE y determinar las acciones para la solución de problemas. La ejecución de las acciones se delega en el plano de acción. En este trabajo se plantea un sistema de gestión muy similar al planteado por nosotros. Sin embargo, en nuestro sistema ha sido incluido un método inductivo de predicción que pensamos es indispensable para llevar a cabo este planteamiento. En dicho trabajo se está investigando la viabilidad de implementar este sistema utilizando redes neuronales.

Hay propuestas de gestión recientes como la propuesta en (Agboma y Liotta 2012) cuya metodología de gestión de la QoE se basa en técnicas de modelado estadístico. Se determinó el grado de influencia de los parámetros de calidad de servicio a nivel de aplicación (Application-level Quality of Service (AQoS)) en la percepción del usuario. Mediante el desarrollo de modelos de predicción se pretende correlacionar los parámetros de calidad de servicio con las estimaciones de las percepciones de la QoE. Adaptados en estrategias de gestión de la QoE podrían reducir el ancho de banda necesario a nivel de aplicación. Con este enfoque se puede lograr un ahorro de recursos significativo, en contraste con las soluciones convencionales de QoS. Coincidimos con este trabajo en el planteamiento del sistema de predicción de la QoE. El problema principal de este trabajo es que el sistema de gestión está diseñado para dispositivos muy concretos como son Personal Digital Assistant (PDA), teléfonos móviles y portátiles.

Existen trabajos con planteamientos mucho menos sofisticados que los anteriores. En (Simoens y col. 2008), los autores centran la solución de problemas en la QoE mediante la gestión de los componentes de la red de acceso. Básicamente, cuando se detecta que un parámetro de la red supera un cierto umbral, se determina cuál es la acción más apropiada para que de forma autónoma pueda ser restaurado. Como vemos, a diferencia de nuestro sistema de gestión, este sistema no tiene la capacidad de aprendizaje de las percepciones de los usuarios.

Capítulo 3

Conceptos generales

En este capítulo se tratarán conceptos generales que serán utilizados en el desarrollo de la memoria. En la sección 3.1 introduciremos conceptualmente los términos QoE y QoS. En la sección 3.2 explicaremos la métrica MOS utilizada para evaluar la QoE y veremos las diferencias con la calidad de vídeo (Video Quality (VQ)). En la sección 3.3 veremos los métodos y las métricas utilizadas para evaluar la VQ. En la sección 3.4 veremos cuáles son los parámetros utilizados para medir tanto la QoE como la QoS. En la sección 3.5 describiremos la infraestructura general de red para un servicio de IPTV. En la sección 3.6 explicaremos brevemente cómo se lleva a cabo una solicitud de servicio de IPTV en el entorno propuesto. En la sección 3.7 describiremos el algoritmo global de gestión para garantizar la QoE a partir de la arquitectura de red planteada. En la sección 3.8 veremos cual es proceso generalizado para el análisis de la QoE en el sistema de gestión. En la sección 3.9 describiremos el banco de pruebas para el estudio de la QoE que utilizaremos en los diferentes estudios realizados en el trabajo de investigación. Por último, en la sección 3.10 veremos algunas propuestas para la solución de problemas en una red de IPTV.

3.1 Introducción a QoE y QoS.

Por un lado, la QoE puede ser definida como las características de las sensaciones, percepciones y opiniones de los usuarios a medida que interactúan con su entorno (ETSI/TR/102 643 2010). Estas características pueden ser desde agradables y divertidas a desagradables y frustrantes. Hay varios aspectos que contribuyen a la QoE del usuario de un dispositivo o servicio multimedia IPTV como son: su idoneidad, eficacia, facilidad de aprendizaje y fiabilidad. La QoE puede significar cosas diferentes para diferentes aplicaciones. Por ejemplo, para una solicitud de

audio, la QoE podría estar relacionada con la fidelidad del sonido, mientras que la QoE de una reproducción de vídeo remota podría estar relacionada con la claridad de la imagen del vídeo recibido. Además, en un mismo medio de comunicación, la QoE no se ve afectada por igual. Por ejemplo, un retardo alto no afecta a la QoE en una transmisión de vídeo de un servicio de vídeo bajo demanda pero puede tener un gran efecto en aplicaciones interactivas como las teleconferencias.

Para garantizar la QoE en la entrega de un nuevo servicio hay que tomar en cuenta según la (ITU-T/P.800 1996): el proceso de inicio de sesión, cómo opera el servicio una vez establecida la sesión y la finalización de la sesión. En este caso la QoE se evaluará de acuerdo a: el esfuerzo requerido por el usuario para llevar a cabo la sesión, el grado de respuesta de una aplicación o servicio, la fidelidad de la información o contenido transmitido así como la seguridad, confiabilidad, entre otros. Si el servicio no está basado en sesiones, la QoE se limitará solamente a los aspectos de interacción con el servicio. Con el fin de medir la QoE en un servicio de flujo multimedia es necesario tener en cuenta las siguientes capas del modelo OSI:

- Capa de aplicación: varios son los parámetros configurables en esta capa que pueden influir en la QoE del usuario final como por ejemplo: la resolución del medio, el tipo de códec, el bitrate, los mecanismos de corrección de errores, etc.
- Capa de Servicio: capa expuesta al usuario, donde se mide la QoE.
- Capa de Transporte: a nivel de la capa de transporte pueden darse problemas en parámetros como retardo, jitter, etc. Valores deficientes de estos parámetros influirán negativamente en la QoE. En esta capa se emplean mecanismos de corrección de errores y de QoS.

Entre los efectos que pueden percibir los usuarios finales por una mala calidad de experiencia según el tipo de flujo multimedia están según la (ETSI/TR/102 643 2010):

- Voz: interrupción, incompreensión, repetición de un efecto de entrecortado, desconexión de las llamadas, etc.
- Vídeo: movimientos erráticos, audio fuera de sincronismo, imágenes ralentizadas o con mala calidad, etc.
- Datos: tiempos variables de respuesta, pantallas que no terminan de abrirse, etc.

Por otro lado, la QoS se refiere a los requisitos necesarios para garantizar las transmisiones en redes tanto de telefonía como de ordenadores. Estos requisitos

se basan en parámetros como: jitter, retardo y paquetes perdidos. En este sentido, se ha desarrollado una gran cantidad de tecnologías para permitir que las redes de computadores lleguen a ser tan útiles como las redes telefónicas. La QoS en telefonía, según la UIT y documentado en la (ITU-T/E.800 2008), se basa principalmente en: el tiempo de respuesta del servicio, las pérdidas, la relación señal-ruido, el habla cruzada, el eco, las interrupciones, la respuesta en frecuencia y los niveles de sonoridad. En las redes de computadores la QoS se basa en: la frecuencia de bits, el retardo, la fluctuación, el jitter, la probabilidad de pérdida de paquetes y/o tasa de errores de bits requeridos, etc. La garantía de la QoS es importante si la capacidad de la red es insuficiente, sobre todo para aplicaciones en tiempo real de flujo multimedia tales como VOIP, juegos en línea e IPTV ya que a menudo requieren una elevada tasa de bits fija y son sensibles al retardo. Este es un concepto que las redes NGN tienen muy en cuenta, sobre todo porque el ancho de banda disponible en las redes actuales es un recurso limitado.

La QoS puede ser proporcionada de dos formas distintas: QoS garantizada o QoS no garantizada (hard - soft QoS) tal como en (Transition networks 2003) han documentado en uno de sus estudios. La QoS garantizada se puede asociar con servicios integrados (Integrated Service (IntServ)) mientras que la QoS no garantizada con servicios diferenciados (Differentiated Service (DiffServ)). Luego, se puede decir que en los servicios integrados se garantiza la QoS de forma absoluta mientras que en los servicios diferenciados se realiza el máximo esfuerzo para garantizarlo. En ambos métodos se garantiza el servicio a aquellos paquetes prioritarios con respecto de los que no lo son. Los dos propósitos fundamentales de la QoS en lo referente a la voz sobre IP y a las aplicaciones de vídeo son: reducir la probabilidad de pérdida de paquetes y permitir que ambas aplicaciones puedan operar en la red al mismo tiempo que otras.

Es común que se confundan las terminologías QoE y QoS(ETSI/TR/102 643 2010); sin embargo, hay una distinción clave sobre ambos conceptos que se basa en la perspectiva desde la que se analiza cada uno de ellos: desde el punto de vista del usuario final para la QoE y desde el punto de vista de las prestaciones de la red para la QoS. La QoE es una medida de extremo a extremo de las prestaciones de servicio del sistema, realizada a nivel de servicio y desde la perspectiva del usuario. Además indica el grado en que el sistema satisface las necesidades del usuario a nivel de calidad. La unidad de medida de la QoE es el valor de percepción/opinión medio MOS. Por otro lado, la QoS es una medida del rendimiento de la capa de red y desde la perspectiva de red. Los parámetros más comunes que se utilizan para medir la QoS son: la pérdida de paquetes, el retardo, el jitter, el ancho de banda y el rendimiento o throughput. El término QoS también hace referencia al conjunto de tecnologías que administran la congestión durante el funcionamiento de una aplicación de red, como por ejemplo una videoconferencia. Proporcionando así diferentes servicios para determinados tipos de tráfico o para determinados grupos de usuarios.

3.2 VQ y MOS

La QoE la podemos definir como el rendimiento global de un sistema desde el punto de vista de los usuarios (ITU-T/FG.IPTV 2007). Los usuarios tienen sensaciones, percepciones y opiniones cuando interactúan con servicios como por ejemplo IPTV. Por otra parte, QoE es un indicador de lo bien que el sistema cumple con las necesidades del usuario y se cuantifica en términos de puntuación de la opinión (o percepción) media o MOS. La escala MOS se define en la recomendación de Unión Internacional de Telecomunicaciones (ITU-T/BT.500 2012) como un mecanismo que sirve como métrica para medir la QoE en IPTV, el cual proporciona una medida subjetiva que cuantifica el impacto que tiene en el usuario la presencia de fallos en el servicio. Estos fallos pueden ser determinados por otras métricas de QoE como la duración de los fallos en el servicio, errores por segundo, y segundos sin disponibilidad del servicio. La UIT presenta un marco de medición de calidad de experiencia que puede dividirse en diferentes puntos de medición específicos: las mediciones obtenidas antes de la transmisión, en la misma red y en el dispositivo receptor.

El MOS es una medida que se ha utilizado durante décadas en las redes de telefonía para obtener la calidad de la red bajo el punto de vista del usuario. Según la recomendación (ITU-T/P.800 1996), MOS es una medida subjetiva, que se utiliza para puntuar la calidad de una llamada y cuya evaluación depende de lo que perciben los evaluadores u oyentes al estar en una habitación aislada de ruidos o factores distorsionadores externos. La medición del MOS para VoIP puede ser más objetiva si nos basamos en cálculos del rendimiento de la red IP. Dichos cálculos se definen en la recomendación (ITU-T/P.862 2001). Como la mayoría de las normas, su aplicación está parcialmente abierta a la interpretación por parte del equipo o fabricante que hace uso de ella.

Tanto para VOIP como para IPTV, el MOS proporciona una valoración numérica de la calidad percibida desde la perspectiva del usuario. El MOS se expresa como un valor comprendido en el intervalo del 1 al 5, donde 1 es la percepción de calidad de audio/vídeo más baja y 5 la más alta. En la tabla 3.1 se muestra el valor del MOS en base a la calidad de audio/vídeo percibida.

Existen ecuaciones para estimar el MOS de forma objetiva, como la que presentan en (Yamamoto y Beerends 1997) para VOIP, basadas en el porcentaje de pérdidas de paquetes y del tamaño de los paquetes tal como se muestra en la ecuación (3.1):

$$Predicted_{MOS} = 4,0 - 0,7 \ln(\text{pérdidas}) - 0,1 \ln(\text{tamaño}) \quad (3.1)$$

donde, *pérdidas* es la tasa de pérdidas de datos en porcentajes desde 0,5% a 50% y *tamaño* es el tamaño de paquete en *ms* con valores desde 3 a 96*ms*. Valores mayores que 4,0 y menores que 1,0 se ajustan a 4,0 y 1,0 respectivamente.

Tabla 3.1: Comparación entre calidad subjetiva y VQM

Percepción	MOS	Deterioro de la calidad
Excelente	5	Imperceptible
Bueno	4	Perceptible pero no molesto
Razonable	3	Ligeramente molesto
Pobre	2	Un poco molesto
Malo	1	Muy molesto

La evaluación con métodos objetivos de la QoE es un tema de gran interés dentro de este ámbito de investigación. En este trabajo presentamos un método inductivo para la predicción del MOS, cuyo análisis teórico se ha llevado a cabo en la sección 5.2 y el práctico en la sección 6.4.

La métrica de calidad de vídeo (Video Quality Metric (VQM)) ha sido desarrollada por el grupo de investigación e ingeniería *International Telecommunications Society* (ITS) de la *National Telecommunications and Information Administration* (NTIA) para proporcionar una medición objetiva de la calidad de vídeo percibida. Con esta métrica se evalúan efectos de percepción tales como: la imagen borrosa, los movimientos espasmódicos o artificiales, el ruido global, el bloque y la distorsión de color. La combinación de todos estos indicadores dan como resultado un único valor de calidad. Se han realizado varios estudios, tal como se muestra en (Wang 2006), en los que se demuestra que la VQM tiene una alta correlación con la valoración subjetiva de la calidad de vídeo. Esta medida ha sido adoptada por el *American National Standards Institute* (ANSI) como un estándar para la obtención de la calidad de vídeo objetiva. Para el cálculo de la VQM se toma el vídeo original y el procesado y se realizan los siguientes pasos:

1. **Calibración:** Se prepara el vídeo para la extracción de características. Se estima el desplazamiento espacial y temporal así como el contraste y brillo del vídeo procesado con respecto al original.
2. **Extracción de características de calidad:** Se extraen un conjunto de características de calidad que caracterizan la percepción del vídeo como son: propiedades espaciales, temporales y de crominancia. Para la extracción de estas características se divide las secuencias de vídeo en subregiones espacio-temporales utilizando funciones matemáticas.
3. **Cálculo de los parámetros de calidad:** Se calculan un conjunto de parámetros que describen cambios en la percepción de la calidad del vídeo

comparando las características extraídas del vídeo procesado con las extraídas del vídeo original.

4. **Cálculo VQM:** VQM se calcula utilizando una combinación lineal de los parámetros calculados en el paso anterior. La VQM puede ser calculada utilizando varios modelos basados en ciertos criterios de optimización. Por ejemplo, el modelo general utiliza una combinación lineal de siete parámetros. Cuatro parámetros se basan en las características extraídas de gradientes espaciales de la componente de luminancia Y, dos parámetros se basan en características extraídas a partir del vector formado por las dos componentes de crominancia (CB, CR) y un parámetro se basa en el contraste de las características temporales extraídas de la componente de luminancia Y.

A continuación mostraremos una posible manera de verificar la correlación entre la calidad subjetiva y la VQM obteniendo valores de MOS a partir de mediciones de VQM. Para ampliar la información se puede consultar (Lloret y col. 2011b). Para llevar a cabo este experimento simulamos una transmisión de flujo de vídeo IPTV con deficiencias en las condiciones de red. El banco de pruebas que hemos utilizado tiene una configuración básica según explicaremos en la sección 3.9. Para calcular la VQM utilizamos el programa *Elecard Video Quality Estimator*. Para ello, le pasamos al programa la secuencia de vídeo antes de la transmisión y la secuencia después de la transmisión. De toda la secuencia seleccionamos dos imágenes para analizar el resultado en base a la calidad recibida. El resultado obtenido se puede observar en la figura 3.1.

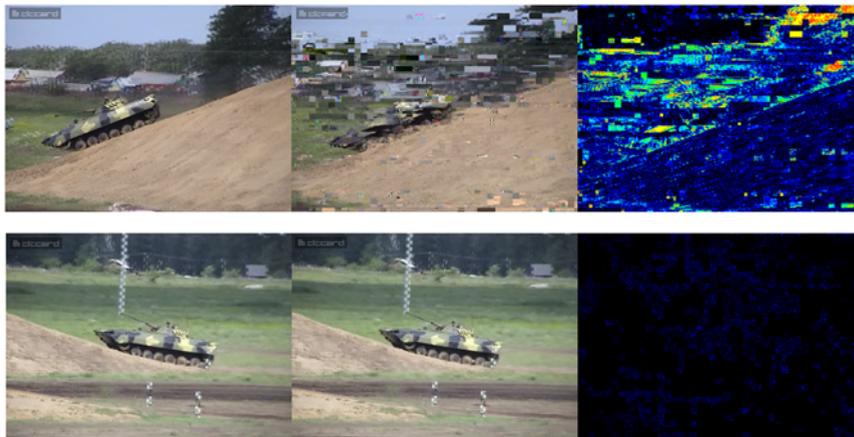


Figura 3.1: Comparación entre el vídeo emitido y recibido para obtener el MOS

En la figura 3.1 podemos observar, de izquierda a derecha, la imagen transmitida en el servidor, la imagen captada en la pantalla del usuario final y la imagen

resultante de la comparación de calidad entre ambas. Cada imagen ha sido captada en un estado de red diferente. La primera fila se corresponde con una secuencia de vídeo transmitida a través de una red con un ancho de banda limitado a 3 Mbps. Como se puede observar en la imagen recibida la calidad es muy baja debido al gran número de paquetes perdidos en dicha transmisión. La imagen resultante del análisis comparativo entre la imagen emitida y la recibida muestra mucha variedad colores y mucha pixelación, lo que indica que la calidad es deficiente. La segunda fila se corresponde con una secuencia de vídeo transmitida a través de una red con jitter variable. La calidad del vídeo recibido es buena como podemos observar en la imagen resultante del análisis comparativo ya que prácticamente no aparece ni variación de colores ni pixelación. Una calidad excelente sería si la imagen del análisis comparativo fuera totalmente negra, ya que querría decir que no hay ninguna diferencia entre la secuencia del vídeo transmitido y el recibido.

Como hemos comentado anteriormente el MOS comprende valores entre 1 y 5, donde 5 es el mejor resultado posible y 1 el peor. Los valores obtenidos en el experimento anterior están en valores de VQM comprendidos entre 0,5 y 4. Para correlacionar el MOS con el VQM asociamos los valores inferiores a 0,5 de VQM a un MOS equivalente a 5, es decir sí $VQM \leq 0,5 \rightarrow MOS = 5$. De la misma forma una $VQM > 4$ lo correlacionamos a un $MOS = 1$. El resto de valores se correlacionaron como: $MOS = 4$ cuando $0,5 < VQM \leq 1,25$, $MOS = 3$ cuando $1,25 < VQM \leq 2,5$, $MOS = 2$ cuando $2,5 < VQM \leq 4$ y $MOS = 1$ cuando $VQM > 4$. Debemos notar que el MOS y la VQM son inversamente proporcionales. Con esto hemos demostrado que el MOS y el VQM son parámetros estrechamente relacionados y que se pueden correlacionar.

3.3 Métodos subjetivos, objetivos e híbridos para el cálculo de la VQ

Hay dos métodos principalmente para medir la calidad de un flujo de vídeo a nivel de usuario: los métodos objetivos y los subjetivos según (ITU-T/BT.500 2012) y (ITU-T/J.144 2004). Los métodos objetivos se basan en medidas objetivas de evaluación de la calidad de vídeo. Los métodos subjetivos se basan en percepciones por parte del usuario durante la visualización del vídeo. La combinación de ambos métodos da como resultado nuevos métodos de medición de la QoE denominados métodos híbridos.

3.3.1 Métodos subjetivos

La calidad percibida de vídeo es por definición un concepto subjetivo ya que depende del evaluador. Existen dos clases de medidas subjetivas: las cualitativas y las cuantitativas. Las cualitativas apuntan a entender los factores sociológicos que determinan la calidad, mientras que las cuantitativas intentan dar una ponderación de la calidad utilizando alguna escala numérica MOS. El método subjetivo de evaluación de calidad de un flujo de vídeo se basa en la combinación de ambas. Para ello se recomienda realizar los siguientes pasos:

1. Primero se eligen una serie de secuencias de vídeo. Estas secuencias deben ser representativas de la realidad que se pretende medir. Por ejemplo, si se quiere determinar el impacto de las pérdidas de paquetes en la red, las secuencias deben presentar casos de pérdidas típicas.
2. Se le solicita a un conjunto de observadores que evalúen las secuencias utilizando alguna escala numérica. Por ejemplo, la valoración de 1 a 5 para la evaluación de la calidad tomando como mala la valoración con el número 1 y excelente la valoración con el número 5.
3. Utilizando criterios estadísticos, se eliminan observadores que por falta de atención (u otros factores externos) no son consistentes con el grupo.
4. Por último, se determina el valor de calidad subjetiva para cada vídeo a partir del valor medio evaluado por los observadores consistentes. A este valor, como ya hemos comentado, se le llama MOS, y puede ser considerado como definición de la QoE.

El proceso de medida subjetiva tiene la ventaja de que es, por definición, la medida más precisa de la calidad de la experiencia. Sin embargo presenta grandes desventajas, por ejemplo, requiere mucho tiempo para ejecutarse y es costosa en recursos humanos, provocando que sea de difícil repetición e imposible de utilizar en un proceso automático.

Los métodos de evaluación subjetiva de vídeos 2D están bastante estandarizados, pero en 3D hay pocos trabajos. En (Wang 2013) se han descrito los métodos de evaluación más comunes. Estos son: Double Stimulus Impairment Scale (DSIS), de la (ITU-T/BT.500 2012), Double Stimulus Continuous Quality Scale (DSCQS), de la (ITU-T/BT.500 2012), y el Absolute Category Rating (ACR), de la (ITU-T/P.910 2008). La diferencia entre el primer y el segundo método es que en el primer método primero se muestra al evaluador el vídeo de referencia y después el distorsionado. A continuación es evaluado. Mientras que en el segundo método, ambos vídeos son mostrados a la vez. La diferencia entre el tercer método y los otros dos es que en éste el vídeo de referencia no se muestra. Únicamente el vídeo distorsionado es evaluado. En (Seuntiëns 2006), hay una clasificación similar basada únicamente en estímulos. En este caso, los métodos de evaluación son divididos

en: doble estímulo (Double Stimulus (DS)), único estímulo (Single Stimulus (SS)) y comparación de estímulos (Stimulus Comparison (SC)). En (Wang 2013) describen dos métodos de evaluación para 3D. El nombre de estos métodos es: Kaptein y Engeldrum. Kaptein se basa exclusivamente en atributos o características de calidad 3D y Engeldrum a parte toma en cuenta características o atributos 2D. Existen otros tipos de clasificación como en (Meesters, IJsselsteijn y Seuntjens 2004), en el que consideran como únicos métodos de evaluación subjetiva: ACR y Paired Comparisons (PC). En (Bosc y col. 2011) plantean un método alternativo a DS, SS y SC, llamado: Single-Stimulus Continuous-Quality Evaluation (SSCQE). Este método se basa en una recopilación continua de evaluaciones de calidad de secuencias de vídeo de tiempo variable.

3.3.2 Métodos objetivos

Los métodos objetivos surgen como un paliativo al problema de los métodos subjetivos comentados en el punto anterior. Por definición, los métodos objetivos son algoritmos o fórmulas matemáticas que dan como resultado medidas de calidad. Existen varios métodos de evaluación objetiva con el fin de medir la calidad de un flujo de vídeo. La utilización de uno u otro de estos métodos depende principalmente de si se dispone del vídeo de referencia o no. Si se dispone del vídeo origen, el método objetivo utilizado se basa en la comparación fotograma a fotograma entre el vídeo origen y el distorsionado. En el caso de que no se disponga, como ocurre en transmisiones de flujo multimedia en tiempo real, el método utilizado ha de basarse en otro tipo de información que pueda ser obtenido en tiempo real, como la señal recibida, el estado de la red, tiempo de zapping, etc... De acuerdo con esta división, las mediciones objetivas de calidad de vídeo se pueden realizar según los siguientes métodos:

Referencia Completa (Full Reference (FR)): El vídeo de entrada del sistema, tras ser procesado, se compara con el de salida. En general, la comparación se hace fotograma a fotograma requiriendo una alineación precisa de las dos secuencias de vídeo. El hecho de que existan retardos variables en el sistema se puede convertir en un verdadero problema para conseguir dicha alineación. Esto ocurre generalmente en las redes de paquetes o transmisiones de flujo multimedia. En la recomendación (ITU-T/J.143 2000), se explican diferentes técnicas de medición objetiva de la percepción de la calidad vídeo en televisión por cable en presencia de una referencia completa. Dichas técnicas se basan en la comparación de cada uno de los frames de transmisión con los de recepción mediante el gradiente de Sobel.

Referencia reducida (Reduced Reference (RR)): Este método está diseñado para predecir la calidad de imagen solamente con información parcial de la imagen de referencia. Se extraen algunos parámetros en el origen y se analizan a la salida del sistema junto a la imagen distorsionada. Este método

puede ser útil en numerosas aplicaciones como por ejemplo, en el sistema de comunicación de vídeo en tiempo real para predecir degradaciones en la calidad de las imágenes y controlar los recursos del flujo. En la subsección 4.3.1 planteamos un método objetivo basado en RR para estimar la VQ, en el que se utiliza información parcial del vídeo a través de un canal auxiliar de datos. La VQ viene dada según la ecuación (4.2).

Sin referencia (Not Reference (NR)): En este caso, sólo la señal recibida de vídeo se utiliza para determinar la calidad de vídeo. También se conocen como técnicas de **single ended**.

Las principales formas de medida o algoritmos utilizados en los métodos objetivos son:

- Peak Signal to Noise Ratio (PSNR)
- Video Quality Metric (VQM)
- Moving Picture Quality Metric (MPQM)
- Color Moving Picture Quality Metric (CMPQM)
- Normalization Video Fidelity Metric (NVFM)
- Moving Image VideoGraphy (MIV), utilizada en Evalvid

Una de las formas de medida más extendida, basada en datos, es el PSNR, que se encarga de analizar, pixel a pixel, los cambios de cuadros entre la señal recibida y una señal de referencia. El problema es que, como se ha demostrado en (Huynh-Thu y Ghanbari 2008), el PSNR no posee una perfecta correlación con la calidad visual percibida. Una medida estrechamente relacionada con el PSNR es el MIV. Con (*Evalvid*) se calcula el MOS de cada fotograma de vídeo recibido mediante un simple mapeo de la PSNR a la escala de MOS. Con el MIV se calcula el porcentaje de fotogramas con un MOS peor que la de referencia (la del vídeo codificado pero sin pasar por el sistema de transmisión) en un intervalo deslizante de fotogramas. Se trata de una medida de calidad que únicamente se encarga de suavizar la respuesta escalonada que ofrece el mapeo del MOS.

Cada vez se hace más necesario el desarrollo de métodos de predicción de la QoE ya que como bien se justifica en (ETSI/TR/102 643 2010) y en (Brooks y Hestnes 2010) los métodos de evaluación subjetiva requieren de un despliegue significativo de recursos lo que provoca un importante retardo en el avance científico en este campo. Es tan importante el hecho de encontrar un método objetivo de evaluación de la QoE que en (Gross y col. 2004) presentan una posible conversión del PSNR a MOS tal como vemos en la tabla 3.2.

Tabla 3.2: Posible conversión de PSNR a MOS

PSNR (<i>dB</i>)	MOS	Percepción
37	5	Excelente
31 - 37	4	Buena
25 - 31	3	Regular
20 - 25	2	Pobre
<20	1	Mala

Con la aparición de nuevos formatos de codificación como los 3D/ST, han ido apareciendo nuevas métricas para la evaluación de la calidad de vídeo como:

- Universal Quality Index (UQI)
- Peak Signal to Noise Ratio - Human Visual System (PSNR-HVS)
- Single-scale Structural SIMilarity (SSIM)
- Visual Signal-to-Noise Ratio (VSNR)
- Weighted Signal-to-Noise Ratio (WSNR)
- Visual Information Fidelity (VIF)
- Information Fidelity Criterion (IFC)
- Noise quality measure (NQM)

Como conclusión, decir que los métodos objetivos son extremadamente útiles para el control de calidad en servicios de red. Funcionalmente pueden ser utilizados para evaluar el rendimiento de procesos como la codificación y cuantificación entre otros.

3.3.3 Métodos híbridos

Ambos métodos, objetivos y subjetivos, presentan inconvenientes importantes al ser aplicados. Los métodos subjetivos son muy costosos y no pueden ser utilizados en procesos automáticos. Los métodos objetivos, para que sean altamente fiables, requieren del vídeo original y el distorsionado simultáneamente, impidiendo así su utilización en sistemas en tiempo real. Por estos motivos surge la necesidad de la

utilización de los llamados métodos híbridos. Estos métodos consisten básicamente en la utilización de herramientas de inteligencia artificial en cuyo aprendizaje se utilizan tanto medidas subjetivas como objetivas. Uno de estos métodos es el conocido como Pseudo-Subjective Quality Assessment (PSQA) basado en redes neuronales y llevado a cabo por el *Project-Team DIONYSOS*. Los métodos híbridos combinan las ventajas de automatización de los métodos objetivos con la precisión de las medidas subjetivas, pudiendo ser aplicados en muchos contextos.

En conclusión, es importante tener presente los factores que afectan a la calidad de experiencia para cada tipo de servicio. Cada servicio debe de ser tratado según el entorno (arquitectura de red) y contexto (protocolos, aplicaciones, etc.) de uso con el fin de garantizar la QoE del usuario final. Es necesario evaluar cada servicio de acuerdo a las expectativas del usuario final, quien será más o menos tolerante dependiendo de la aplicación o servicio (vídeo, voz y datos) que utilice. Un buen diseño de red que garantice una QoS elevada puede asegurar una alta QoE. Sin embargo, esto nos puede llevar a sobredimensionar los recursos necesarios con el consecuente aumento de la inversión y el encarecimiento del producto final. Por tanto, debe existir un equilibrio entre QoE y QoS y los costes en la creación de servicios multimedia.

3.4 Parametrización general de QoE y QoS

En esta sección analizaremos los principales parámetros de medida que se utilizan para evaluar la QoE y QoS de una red de IPTV. Cabe recalcar que aunque la QoE y la QoS son términos diferentes, están estrechamente relacionados tal como se ha explicado en la sección 3.1. Ambos términos hacen referencia a la calidad del servicio entregada al usuario, pero evaluándola bajo dos puntos de vista distintos: la QoE desde el punto de vista del usuario y la QoS desde la tecnología necesaria para la entrega de dicho servicio, es decir las redes de telecomunicaciones. Por tanto los parámetros necesarios para garantizar la QoS dependerán de la QoE y viceversa.

3.4.1 Parámetros de medida de la QoS

Los principales parámetros de QoS para IPTV según la (ITU-T/E.800 2008) son:

La degradación media: Degradación de la QoE debido al códec utilizado en una transmisión de voz o vídeo.

Retardo: El tiempo medio de ida durante la transmisión de un paquete que se envía a través de la red desde el emisor al receptor y viceversa. Si los paquetes son de vídeo, definiremos el retardo como el tiempo necesario por la red de

transporte para entregar dichos paquetes al usuario final. El retardo se utiliza en el cálculo del rendimiento extremo a extremo.

Jitter: La variación en el tiempo de retardo de los paquetes que llegan a su destino. Los paquetes se envían a intervalos regulares desde el emisor al receptor, pero debido a la latencia de la red el intervalo entre los paquetes pueden variar en el destino. Esta variación puede afectar a la calidad de los medios de comunicación.

Promedio de degradación por jitter: Degradación de la QoE causada por fluctuaciones en la red durante la transmisión. Degradación por fluctuación media es la cantidad de fluctuación que ha contribuido a la degradación media. Mediante este valor se puede determinar si el jitter es el principal contribuyente en la red de la degradación de la QoE.

La pérdida de paquetes: Relación entre los paquetes perdidos y el número total de paquetes enviados. La pérdida de paquetes influye directamente en la QoE y en características como el rendimiento de la transmisión.

Promedio de degradación por paquetes perdidos: Degradación de la QoE causada por la pérdida de paquetes durante la llamada. Degradación por pérdida de paquetes media es la cantidad de pérdida de paquetes que ha contribuido a la degradación media. Mediante este valor se puede determinar si la pérdida de paquetes era el principal contribuyente de la degradación de la QoE.

Velocidad de bits de vídeo: Tasa media, en bits por segundo, del proceso de codificación por cada imagen de un vídeo. La tasa de bits de vídeo se utiliza en los contadores de rendimiento.

Pérdida de frames de vídeo: Número medio de imágenes consecutivas, o fotogramas (frames) de vídeo, perdidas durante la transmisión. Debido a que las imágenes de vídeo pueden abarcar varios paquetes, este valor puede ser más útil que la pérdida de paquetes en la evaluación de la calidad de vídeo.

Tiempo de zapping: En transmisiones de vídeo es el tiempo que necesita un abonado para cambiar de canal y recibir el vídeo en un nuevo canal. Un retraso de cambio de canal de 1 seg. se considera un zapping aceptable. Se considera instantáneo entre $100ms$ y $200ms$.

Tiempo de retardo Round-Trip (RTT): Tiempo que tarda un paquete de datos enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino.

Tasa de frames: Fotogramas o frames por segundo que se producen en la transmisión.

Estimación de ancho de banda: Ancho de banda estimado disponible en la transmisión.

Densidad de ráfaga: La fracción de paquetes de datos Real-Time Transport Protocol (RTP) RFC 3550 (Schulzrinne y col. 2003), en periodos de ráfaga, perdidos o descartados desde el inicio de la recepción. Un periodo de ráfaga es un periodo en el que son, o bien una alta proporción de paquetes perdidos o eliminados por la llegada tardía de los mismos.

Longitud de ráfaga: La duración media, expresada en milisegundos, de los periodos de ráfaga que se han producido desde el comienzo de la recepción.

Densidad de gap: La fracción de paquetes de datos RTP entre ráfagas desde el inicio de recepción que se perdieron o descartaron.

Espacio entre etiquetas: La duración media, expresada en milisegundos, de los periodos de carencias que se han producido desde el inicio de la recepción.

Hay muchos factores, a parte de los nombrados, que comprometen a la calidad de audio y vídeo tal como hemos podido comprobar en alguno de nuestros trabajos como en (Lloret y col. 2013). Por ejemplo, está la cantidad de suscriptores al servicio de IPTV, su comportamiento y el tiempo de convergencia del servicio, entre otros factores.

Los parámetros de red como el jitter, retardo, paquetes perdidos y tiempo de zapping se pueden medir tanto en la capa de transporte como en la de aplicación de una red TCP/IP. Estos parámetros son muy utilizados para evaluar la QoS pero también para ayudarnos a estimar la QoE como ya hemos visto. Los índices de entrega medios (Media Delivery Index (MDI)) de jitter, retardo y paquetes perdidos son aceptados por la industria para medir la calidad de vídeo y audio a través de la red de distribución. MDI está definido en la RFC4445 (Welch y Clark 2006) y respaldado por la *IP Video Quality Alliance*.

En (Canovas y col. 2010b) determinamos los umbrales de los parámetros de QoS en base a la QoE para una transmisión de flujo de vídeo estándar (Standard Definition Television (SDTV)) en entornos cableados. Tal como se indica en el anterior trabajo, entre el 0,5% y 1,5% de pérdidas de paquetes es aceptable. Para este tipo de transmisiones, según el forum DSL WT-126, se recomienda un máximo de pérdidas de 5 paquetes IP consecutivos cada 30 minutos, y 4 horas para transmisiones HDTV. Por otro lado, el valor del retardo se considera: bueno de 0ms a 150ms, aceptable entre 150ms y 300ms y malo superior a 300ms. En cuanto al valor del jitter se considera: bueno de 0ms a 20ms, aceptable entre 20ms y 50ms y malo si supera los 50ms. Estos resultados pueden ser contrastados en la (ITU-T/E.802 2007). Cabe decir, con respecto al jitter, que si se utilizan técnicas de buffering (e.g., mediante el uso de jitter/playout buffer), estos valores se

pueden compensar, aunque sea a costa de un mayor retardo en las comunicaciones, haciendo que el impacto negativo en la QoE de este parámetro se pueda aislar.

En (Lloret y col. 2011b) y en (García y col. 2009) analizamos el tiempo de zapping para diferentes tecnologías. Los resultados obtenidos del estudio se pueden observar en la figura 3.2.

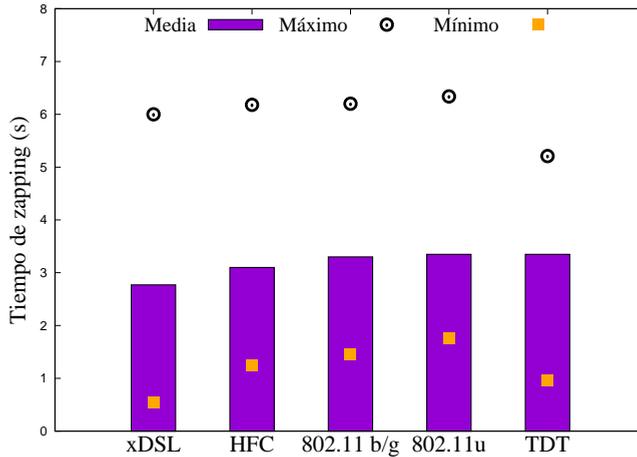


Figura 3.2: Tiempo de zapping para diferentes redes de acceso.

En la figura 3.2 se compara el tiempo de zapping en redes xDSL, HFC, IEEE 802.11 b/g y IEEE 802.3u. Como se puede observar el tiempo de zapping más alto se da en la tecnología IEEE 802.11u y el más bajo en xDSL.

3.4.2 Parámetros de QoE en transmisiones de vídeo 2D

Existen parámetros, a nivel de interacción del usuario con el servicio, que se utilizan como métodos subjetivos de evaluación de la QoE. Según la ETSI/TR/102 643 2010 estos son: la disponibilidad de contenidos, la facilidad y la indexación de contenidos disponibles, el interfaz de usuario, la paleta de colores, la ergonomía, el diseño de la navegación y la guía de programas.

Dentro de los métodos subjetivos de evaluación existen métricas para evaluar la percepción humana sobre la calidad de una imagen o vídeo y que afectan a la QoE. Estas métricas las hemos identificado como: blockiness (pixelado), blur (estela), ringing (ruido/borroso), color bleeding (corrimiento de color), black square blocks (pixels negros), ghosting (sombra), grain lost contrast (zonas de pérdida de con-

traste), pérdida de imágenes, columnas y error de crominancia. En la figura 3.3 mostramos una serie de ejemplos de estas métricas. Durante una transmisión de flujo de vídeo, los factores que suelen provocar estos errores son: la codificación, la compresión, la cuantificación, el bitrate, la resolución o tamaño de imagen, las propias condiciones de red durante la transmisión como: jitter, retardo, paquetes perdidos y ancho de banda así como también el proceso de handover/roaming en caso de transmisiones inalámbricas.

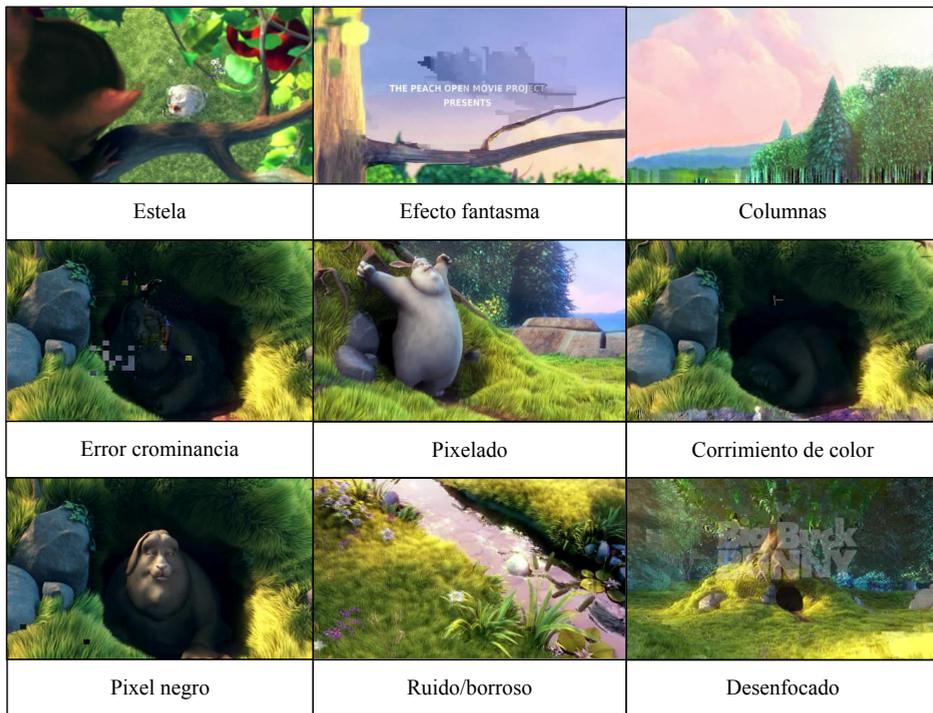


Figura 3.3: Errores perceptibles a nivel de imagen que afectan a la QoE

Como acabamos de comprobar los parámetros de red afectan a la QoE. En la sección 6.1 planteamos una serie de ecuaciones que estiman el valor de la QoE en base a dichos parámetros. Estas ecuaciones las consideraremos por tanto como métodos objetivos para la evaluación de la QoE.

3.4.3 Parámetros de QoE en transmisiones de vídeo 3D/estereoscópica

Según (Meesters, IJsselsteijn y Seuntiens 2004), ni los parámetros de calidad de vídeo ni las métricas utilizadas en 2D son suficientes para analizar la calidad de vídeo 3D/ST. La percepción de la calidad de un vídeo 3D/ST depende de sensaciones tales como: la nitidez, la percepción de profundidad, fatiga ocular, la naturalidad o la presencia de diafonía entre las vistas. Los atributos que afectan a estas sensaciones son según (Meesters, IJsselsteijn y Seuntiens 2004) y (Benoit y col. 2008):

Efecto de distorsión trapezoidal o efecto lápida (keystone distortion):

Es el efecto causado por el intento de proyectar una imagen sobre una superficie en un ángulo. Da la sensación como que la imagen está tumbada.

Curvatura en el plano de profundidad (depth-plane curvature): Al igual que el efecto de distorsión trapezoidal, se produce una distorsión en la imagen.

Efecto de teatro de títeres (puppet theater effect): Al producirse este efecto da la sensación que los objetos 3D están distribuidos como en un teatro de títeres, con poco efecto de profundidad y en dos planos.

Efecto doble (Crosstalk effect): Este efecto se produce en pantallas estereoscópicas y se define como la fuga de la imagen de un ojo en la imagen del otro ojo. Esto hace que la imagen se vea doble.

Efecto de cartón (cardboard effect): Al observar una pantalla estereoscópica, a veces podemos no detectar la estructura 3D de algunos objetos, este fenómeno se denomina efecto de cartón. Este efecto es una distorsión peculiar de las imágenes estereoscópicas y describe el fenómeno por el cual los objetos parecen estar aplanados en profundidad.

Distorsión de cizallamiento o lateral (shear distortion): Efecto como que los objetos están deformados hacia un lado con respecto a un punto de referencia.

Efecto de valla (picket-fence effect): La máscara entre las columnas de píxeles de una pantalla Liquid Crystal Display (LCD) da lugar a la aparición de franjas o bandas verticales en la imagen. Estas franjas son conocidas como efecto valla.

Volteo de la imagen (image flipping): Es una inversión de la imagen con respecto al eje horizontal.

En (Bosc y col. 2011) han añadido otros efectos como: el efecto fantasma (ghosting), el efecto de escalera (staircase) y el efecto parpadeo. El efecto fantasma

se produce por una réplica en la imagen que acaban superponiéndose. El efecto escalera es producto de un pixelado en los bordes o contornos de la imagen. La resolución es otro efecto al que le han dado importancia en (Heynderickx y Kaptein 2009).

Hay otros factores relacionados con las pantallas 3D/ST que pueden influir en la QoE. Las pantallas 3D/ST se pueden clasificar según (Wang 2013) en: estereoscópicas y auto-estereoscópicas. En las **pantallas estereoscópicas** el usuario tiene que utilizar gafas (Kim y col. 2009), sin embargo en las pantallas auto-estereoscópicas no es necesario (Dodgson 2005). Las pantallas estereoscópicas pueden ser de multiplexación temporal o multiplexación espacial. Las pantallas estereoscópicas de multiplexación temporal utilizan gafas activas (active shutter glasses) y las de multiplexación espacial gafas pasivas (passive polarized glasses). Las **pantallas auto-estereoscópicas** son pantallas 3D volumétricas en las que se representan los objetos en 3 dimensiones (Favalora 2005). A partir de esta clasificación hemos extraído una serie de parámetros subjetivos que pueden afectar a la QoE:

Pérdida de resolución: Una pantalla 3D es un sistema multivisión con respecto al observador. Para aumentar el efecto de profundidad se aumentan el número de puntos de vista pero con la consecuencia de que se produce una pérdida de resolución. La razón es que el número de píxeles que se pueden colocar en una pantalla de cristal líquido es limitado. Para paliar el problema se utiliza una tecnología de procesamiento de imágenes llamada matriz de paso de píxeles 3D, pero no deja de afectar a la QoE del usuario final.

Zona de visualización: Las imágenes en pantallas 3D se han diseñado con un ángulo de visión de 62 a 65 grados, lo cual puede ser incómodo si el usuario no ve la imagen desde en frente, desde una distancia determinada y desde un ángulo determinado, ya que el ojo puede detectar una imagen 2D en algunas zonas de la pantalla si su posición no es correcta tal como se ha analizado en (Lambooi, IJsselsteijn y Heynderickx 2007).

Distorsiones: Los sistemas de vídeo estereoscópico buscan mostrar verdaderas vistas 3D de escenas del mundo real. Durante el proceso de adquirir y posteriormente mostrar una imagen en pantalla puede que ocurran distorsiones. Estas distorsiones pueden modificar la percepción del observador de la escena representada o incluso reducir la calidad de la imagen estereoscópica.

Gafas: Otro aspecto importante es la necesidad de gafas para la visualización correcta del 3D/ST. Las gafas son incómodas para el usuario. El usuario está acostumbrado a ver la TV sin la necesidad de ningún tipo de accesorio extra y actualmente el uso de gafas es una restricción para poder ver el 3D en condiciones.

3.5 Infraestructura general de red para IPTV

Las nuevas tecnologías y servicios están fomentando el desarrollo de modelos de negocio para la entrega de televisión a través del protocolo IP. De acuerdo con la (ITU-T/BT.500 2012), el servicio IPTV debe garantizar una calidad de servicio adecuada, seguridad, interactividad y fiabilidad. IPTV debe tener una correcta QoS y una adecuada QoE que satisfaga las necesidades y expectativas de los usuarios finales. Para conseguirlo, los proveedores de IPTV deben tener básicamente una red IP adecuada con el fin de que no se produzcan principalmente interrupciones en el servicio, degradaciones de imagen en recepción o largos tiempos de zapping. En el siguiente apartado describiremos cuál es la arquitectura de red que planteamos con el fin de conseguir una red IP adecuada para la entrega de un servicio de IPTV.

3.5.1 Arquitectura de red para IPTV

Podríamos definir la red de IPTV como un conjunto de redes de banda ancha que son capaces de soportar el ancho de banda requerido para la entrega de vídeo. Como se puede observar en figura 3.4, la topología de red IPTV se puede dividir en 5 partes principalmente: la cabecera de red, la red troncal, la red de distribución, la red de acceso y la red del cliente. Por lo general, en una red de IPTV, las secuencias de flujo de vídeo y audio se empaquetan y se envían utilizando contenedores con formato MPEG para el transporte del flujo (MPEG-Transport Stream (MPEG-TS)) y a través del protocolo RTP (*RFC3550* (Schulzrinne y col. 2003)). En RTP, se deben cumplir las normas específicas de empaquetado para cada códec de vídeo. Al llevar a cabo este proceso tendremos flujos empaquetados RTP (*RFC2250* (D. Hoffman y Civanlar 1998)). El protocolo RTP soporta flujo de vídeo en tiempo real, con mecanismos de control, con el fin de sincronizar flujos de audio/vídeo diferentes. A este protocolo se le denomina Real Time Streaming Protocol (RTSP) (*RFC2326* (Schulzrinne, Rao y Lanphier 1998)). RTSP es un protocolo no orientado a conexión, en lugar de esto, el servidor mantiene una sesión asociada a un identificador. En la mayoría de los casos RTSP usa TCP para datos de control del reproductor y User Datagram Protocol (UDP) para los datos de audio y vídeo. Aunque también se puede utilizar TCP en caso de que sea necesario. En el transcurso de una sesión RTSP, un cliente puede abrir y cerrar varias conexiones con un servidor.

A continuación explicaremos cada una de las partes de la arquitectura de red para servicios de IPTV:

Cabecera de la red: En la cabecera de red se encuentran la granja de servidores y el controlador de vídeo. Estos dispositivos constituyen los componentes básicos de la infraestructura IPTV. La granja de servidores son los encargados de distribuir los contenidos IPTV a la red principal, en calidad de

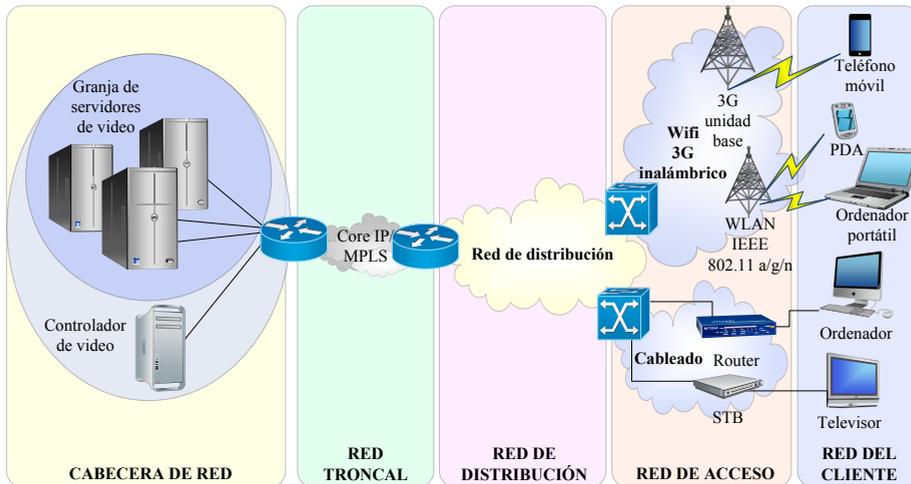


Figura 3.4: Arquitectura general de red para servicios de IPTV

proveedor de servicios, hasta la red de acceso del usuario final. Luego estos dispositivos reciben, transforman y distribuyen el contenido a los suscriptores que lo soliciten. El controlador de vídeo es el encargado de gestionar la distribución de dicho contenido con el fin de garantizar la QoE del usuario. También controla el acceso de aquellos usuarios autorizados. Como vemos la cabecera de red es el punto más crítico de la red IPTV.

Red troncal: Recibe los flujos de vídeo de la cabecera de red y los encamina hasta la red de distribución. Luego, la red troncal interconecta los proveedores de servicios y las aplicaciones de IPTV. Las tecnologías que se utilizan a menudo en la red troncal son: Gigabit Ethernet, SONET/SDH, y las tecnologías xWDM. Las topologías más comunes en esta parte de la red son: punto a punto y anillo. Estas deben ser escalables. Los routers son los dispositivos más importantes de la red troncal de IPTV. Se encargan de la conmutación y enrutamiento de los paquetes entre la cabecera de red y la red de distribución. Por ello, estos dispositivos deben tener un alto rendimiento.

Red de distribución: La red de distribución une la red troncal, a través del router troncal, con la red de acceso, mediante el router de agregación. Luego, la red de distribución de datos debe realizar tareas de transmisión y conmutación de forma eficiente. El router de agregación proporciona tanto

multiservicio y enrutamiento como estabilidad a la red. El hecho de proporcionar multiservicio facilita la transformación eficiente entre los protocolos de red IP y MPLS. Por tanto, podemos decir que la principal función del router de agregación es la de multiplexar diferentes proveedores de servicios y adaptar el sistema de transporte a las características específicas del bucle de abonado. En la red de distribución también existe la posibilidad de que haya servidores de distribución de vídeo. En este caso, la función principal de esta parte de red es la distribución de contenido múltiple de los diferentes proveedores de servicios.

Red de acceso: Permite al usuario conectarse al proveedor de servicios y permite el acceso a los contenidos multimedia. El primer requisito de una red de acceso es tener suficiente ancho de banda para soportar múltiples canales de IPTV para cada suscriptor así como para otros servicios tales como telefonía y datos. El contenido multimedia es transportado al usuario final a través de la red de acceso. Esta red gestiona las exigencias del usuario utilizando canales de retorno. En la actualidad las tecnologías de acceso más utilizadas son xDSL, FTTx o inalámbrica. Estos tipos de tecnologías permiten gran ancho de banda aunque no para usuarios finales con característica de movilidad, éste es un aspecto importante a tener en cuenta.

Red del cliente: La red del cliente permite la comunicación y el intercambio de información entre las computadoras, Set-Top-Box (STB) o dispositivos móviles que se encuentran en esa red y la red de proveedor de servicios. Permite acceder a los recursos disponibles en la red IPTV. El medio compartido en la red del cliente puede ser por cable o de forma inalámbrica, utilizando tecnologías tales como Fast Ethernet, WiFi (IEEE 802.11a/b/g/n), 3G y actualmente 4G.

Para garantizar la QoE y QoS en las diferentes capas de la arquitectura de red de IPTV debemos tener en cuenta los siguientes aspectos: la cabecera de vídeo en red, la red IP, el middleware y el dispositivo cliente.

- La cabecera de la red de vídeo es donde las aplicaciones y el contenido de los programas se almacenan.
- La red IP es el mecanismo de transporte que distribuye el contenido de televisión y servicios interactivos de la cabecera a la casa del usuario.
- El middleware es el software que controla el contenido de televisión y los servicios interactivos de la cabecera de red a la casa del usuario a través de la red.
- El dispositivo cliente se encuentra en la casa del usuario y es mediante el cual se visualiza y escucha los servicios de IPTV.

Cada uno de los aspectos anteriormente nombrados puede afectar a la QoE de una manera diferente. Por ello son considerados, analizados y parametrizados de forma diferente. En un servicio de IPTV, la vigilancia de la calidad de las secuencias de vídeo se tiene que realizar en todas las partes de la red. Es una tarea difícil por dos motivos principalmente: por un lado, los requisitos en una red IPTV deben de ser rigurosos, y por otro lado, es difícil aislar los problemas de IPTV de los de los otros servicios.

3.6 Proceso para la solicitud de un servicio de IPTV

El proceso de solicitud del servicio de IPTV que planteamos para un nuevo cliente es el siguiente: en primer lugar se autentifica el cliente. Mientras el cliente no es autenticado no se le asignará un canal. Una vez el cliente es autenticado se le asignará un canal en función de las características del dispositivo y de la conexión desde la cual se ha realizado la petición. El cliente estará a la espera de recibir el canal asignado para así unirse al grupo multicast correspondiente. Pasado un cierto tiempo y un cierto número de intentos, si el cliente continúa sin recibir el canal asignado se llevará cabo la reanudación del proceso o la desconexión definitiva. El proceso anteriormente descrito se puede observar en la figura 3.5.

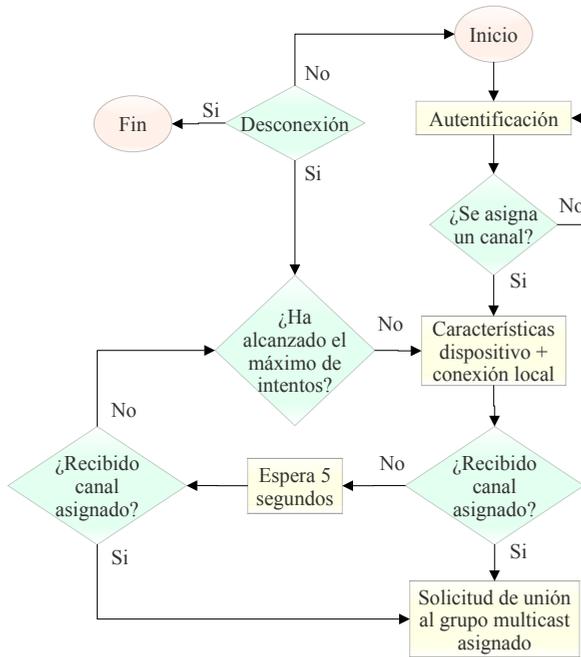


Figura 3.5: Algoritmo de petición de servicio IPTV

3.7 Algoritmo de gestión global

El algoritmo de gestión para la entrega de flujo de vídeo IPTV está diseñado para ser integrado en el controlador de vídeo situado en la cabecera de red según la arquitectura planteada en la subsección 3.5.1. Tal como se explica en dicha sección, la arquitectura dispone de una granja de servidores de flujo de vídeo para la entrega de contenido media, un dispositivo de usuario que decodifique dicho contenido, routers de control y el elemento principal de la arquitectura, el controlador de vídeo. El controlador de vídeo gestiona todos los procesos de entrega, envía información para monitorear la red y gestiona la granja de servidores de flujo de vídeo. El controlador intercambia información con aquellos dispositivos que necesite monitorear. Para la transmisión de vídeo se utiliza los protocolos RTP/RTCP sobre UDP. Una de las cualidades importantes de este protocolo es que nos permite conocer qué secuencias de vídeo han sido enviadas por el servidor.

Este es un aspecto que hemos tenido en cuenta en el desarrollo del sistema de gestión.

Tal como vemos en la figura 3.6 el algoritmo de gestión tiene dos partes claramente diferenciadas. Una es la parte de gestión de solicitudes de canal para IPTV y otra es la parte de gestión de alarmas que se dan cuando existe una degradación de la QoE. Si se activa uno de estos dos eventos se inicia el algoritmo y entra en siete posibles procesos de gestión:

- Gestión de inicio de la conexión, ampliado en sección 3.6.
- Gestión heurística de la transcodificación, ampliado en sección 6.1.
- Gestión inteligente de la transcodificación, ampliado en sección 6.4.
- Gestión de conexiones inalámbricas, ampliado en sección 6.2.
- Gestión de conexiones móviles, ampliado en sección 6.3.
- Gestión de conexiones cableadas, ampliado en sección 6.1.
- Gestión de red, ampliado en sección 3.10.

3.7.1 Activación de un evento de petición de servicio IPTV

El controlador de vídeo está en escucha continuamente a la espera de recibir una solicitud de canal o una alarma. Cuando un nuevo usuario quiere recibir contenido IPTV envía una petición al router de agregación de la red de distribución. Para poder realizar este tipo de peticiones se utilizan routers capaces de abrir flujo de vídeo (openflow capable router) (McKeown y col. 2008) y explicado en (Open Networking Foundation 2012). La petición será aceptada o rechazada en base a los recursos disponibles. Si la petición es rechazada, el router enviará un mensaje al usuario final indicándole que lo intente de nuevo. Si el router acepta la petición, la reenvía junto con los valores de los recursos disponibles al controlador de vídeo.

Al recibir el controlador de vídeo la petición se activará el evento de solicitud de canal, dando lugar al proceso de **gestión de inicio de conexión** (sección 3.6). Al inicio de este proceso, el controlador comprueba la autenticación del usuario. Si no es correcta, la solicitud se rechaza. Si es aceptada, se comprobarán los recursos disponibles en la cabecera de red para la entrega del servicio de IPTV. Si no son suficientes la solicitud será rechazada y el servidor continuará a la escucha. Si hay suficientes recursos se asigna un canal al usuario y se enviará una solicitud pidiéndole las características del dispositivo y las características de la conexión desde dónde se ha solicitado el servicio. Tras recibir esta información se comprueba el tipo de conexión desde la que se va a realizar la petición. Ésta puede ser inalámbrica, cableada o desde dispositivo móvil.

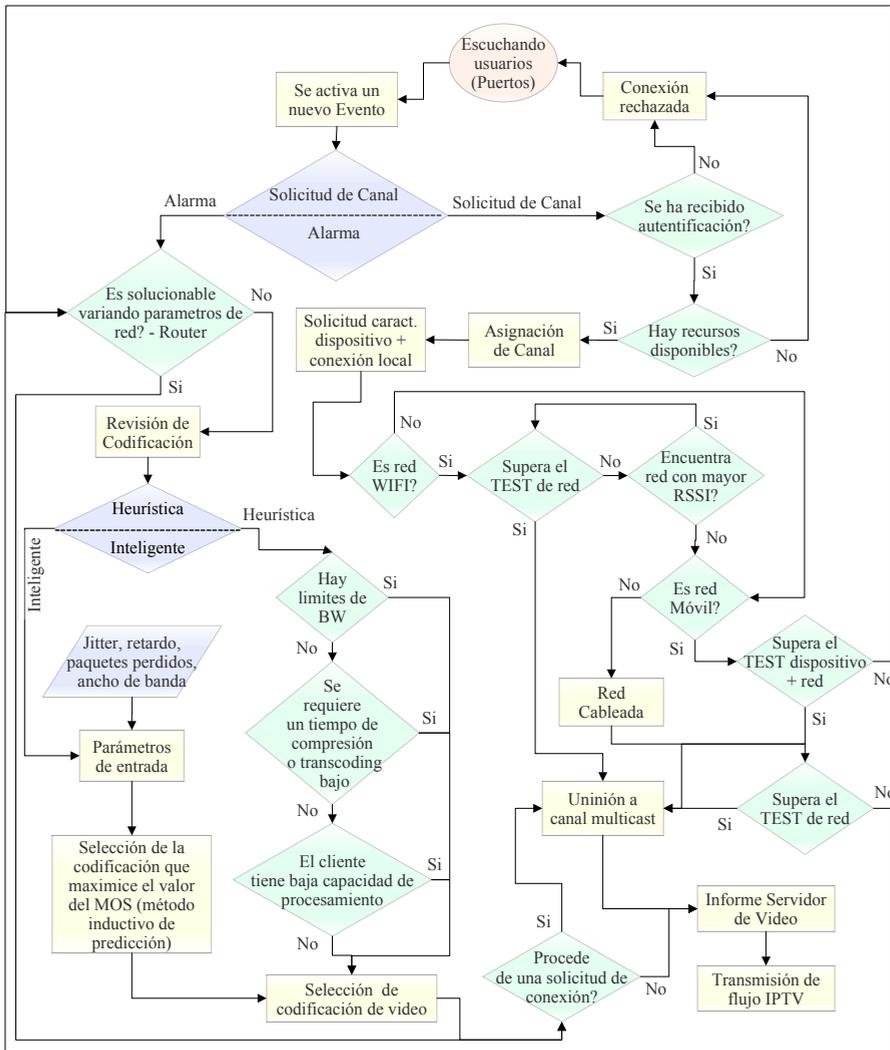


Figura 3.6: Algoritmo de gestión global para servicios de IPTV

En el caso de que sea inalámbrica el controlador inicia el proceso de **gestión de conexiones Inalámbricas** (sección 6.2). Comprueba el estado de la red WIFI con el fin de que se garantice la QoE del usuario final. Si no es posible garantizarlo,

se llevará a cabo una búsqueda de redes próximas en mejores condiciones. Si se localiza alguna, el dispositivo se unirá a ella y se comprobará si en las nuevas condiciones de red se garantiza la QoE. Este proceso se repetirá hasta que esto último se cumpla. Si finalmente se cumple el usuario será unido a un canal multicast y se iniciará la transmisión del flujo de vídeo. Por el contrario, si no se llega a encontrar ninguna red que garantice la QoE se comprobará el tipo de dispositivo desde donde se está realizando la petición. Si el dispositivo es móvil se inicia el proceso de **gestión de conexiones móviles** (sección 6.3). Se analizan las características del dispositivo y las condiciones de red que mejor se ajusten a este dispositivo. Tras analizar toda esta información, si el controlador de vídeo determina que si se puede garantizar la QoE del usuario, éste será unido a un grupo multicast. Si por el contrario, concluye que no es posible garantizarla se deberá pasar a realizar ajustes de red o en su defecto transcoding del vídeo. Lo primero que se comprueba es si es posible variar desde el router más próximo al usuario los parámetros de red que sean necesarios para que así se pueda transmitir el vídeo con garantías de calidad, realizando de esta forma un proceso de **gestión de red** (sección 3.10). Si funciona, se unirá a un grupo multicast y se inicia la transmisión. Si esto tampoco funciona, se llevará a cabo un proceso de transcoding. El proceso de transcoding podrá ser de dos tipos, heurístico o inteligente, dependerá del vídeo y las condiciones de red. Si la seleccionada es heurística se inicia el proceso de **gestión heurística de codificación** (sección 6.1) y se comprueban parámetros como BW, tiempo de compresión, etc. Estos factores determinarán la codificación del vídeo que se va a transmitir. Para ampliar la información sobre el proceso de transcoding se puede consultar la sección 4.4. Si el método seleccionado se basa en un sistema inteligente se iniciará el proceso de **gestión inteligente de codificación** (sección 6.4), basado en la estimación de la QoE. Se comprobará para cada uno de los códecs cuál maximiza el valor estimado de la QoE, obteniendo así la codificación óptima para el contexto en el que nos encontramos. El sistema heurístico suele funcionar bien al inicio de la transmisión ya que todavía la configuración de la transmisión no está establecida. Sin embargo, durante la transmisión es más eficiente utilizar transcoding inteligente. Una vez finalizado este proceso, se une al usuario a un canal multicast, se informa al servidor cuál es el códec de vídeo para la transmisión y se inicia la transmisión del flujo de vídeo.

Si la conexión no es inalámbrica y se realiza desde una red móvil el proceso de gestión que se llevaría a cabo directamente es la **gestión de conexiones móviles** (sección 6.3). Como ya hemos comentado en el párrafo anterior, se analizan las características del dispositivo y las condiciones de red que mejor se ajusten a este dispositivo. En (Bri y col. 2013) se explica el sistema que desarrollamos para obtener las características de los dispositivos de usuario. Si el controlador de vídeo determina que sí se puede garantizar la QoE del usuario, éste será unido a un grupo multicast. Si se concluye que no es posible se pasará a realizar ajustes en la red o en su defecto en la transcoding del vídeo ya sea de forma heurística

o inteligente. Tras finalizar con éxito uno de estos procesos el usuario se unirá a un grupo multicast, se informará al servidor de cuál es el códec de vídeo para la transmisión y se iniciará la transmisión de flujo de vídeo IPTV.

Si la conexión se realiza desde un entorno cableado, el servidor controlador de vídeo comprobará si se puede garantizar la QoE, iniciando el proceso de **gestión de conexiones cableadas** (sección 6.1). Si es así, se une el usuario a un canal multicast y se inicia la transmisión. Si no ocurre esto se inicia un proceso similar al caso anterior pero con la salvedad que para este caso no se necesita conocer las características del dispositivo. El motivo principal es que no existen tantas restricciones a nivel de hardware en estos entornos de red como en los móviles. Aunque en las nuevas generaciones de móviles las limitaciones de hardware están desapareciendo ya que se están desarrollando dispositivos muy potentes. Tras analizar las condiciones de red y parámetros de usuario como el tiempo de zapping, el controlador determina si se puede garantizar la QoE del usuario. Si se puede se une a un canal multicast. Si se concluye que no es posible se pasará a realizar ajustes en la red y si con esto no se soluciona se llevará a cabo la transcodificación del vídeo ya sea de forma heurística o inteligente. Tras finalizar con éxito uno de estos dos procesos el usuario se unirá a un grupo multicast, se informará al servidor de cuál es el códec de vídeo para la transmisión y se iniciará la transmisión de flujo de vídeo IPTV.

3.7.2 Activación de un evento de alarma

Como ya hemos visto en este capítulo, se ha demostrado que parámetros de QoS como pérdida de paquetes, ancho de banda, etc, afectan a la QoE. Además, en la sección 6.1 hemos realizado un estudio que lo demuestra. En base a esta premisa, hemos planteado los siguientes métodos de activación de alarmas.

Existen dos formas para que el sistema de gestión active una alarma. Una es desde el controlador de vídeo y otra desde el router de la red de acceso. El controlador de vídeo se encuentra continuamente en un proceso de monitorización de red y estimación de la QoE mediante métodos inductivos de predicción. Con la monitorización, mediante mensajes Simple Network Management Protocol (SNMP), se solicitan aquellos parámetros que se desean controlar ya sean de red (jitter, BW,...) o de usuario (tiempo de zapping,...). Cuando se estime una degradación importante del valor de la QoE se activa la alarma. Cuando la alarma es activada por el router, es porque se ha sobrepasado unos límites de: retardo max., jitter max., tasa de pérdidas max., BW min., etc. Al sobrepasar estos límites el router lo detecta y envía inmediatamente una señal de alarma al controlador de vídeo haciendo que se active. Cabe decir que la alarma se activará si se repite más de 3 veces en un tiempo mínimo de 30 segundos, para ambos casos. La justificación es el no tomar en cuenta falsas alarmas.

En el caso de que se active el evento de alarma, lo primero que hace el sistema de alarma es comprobar si se puede solucionar variando los correspondientes parámetros de red mediante el proceso de **gestión de red** (sección 3.10). Si se puede solucionar así no se realiza ningún cambio en la transmisión. Si no se pudiera solucionar de esta forma, se llevaría a cabo una transcodificación ya sea heurística o utilizando técnicas de inteligencia artificial. Tras la finalización de la transcodificación se informa al servidor de cuál es el códec de vídeo para la transmisión y se continua con la transmisión de flujo de vídeo IPTV.

Para asimilar conceptualmente todo este proceso global que realiza el sistema de gestión y que acabamos de explicar podemos observar la figura 3.7.

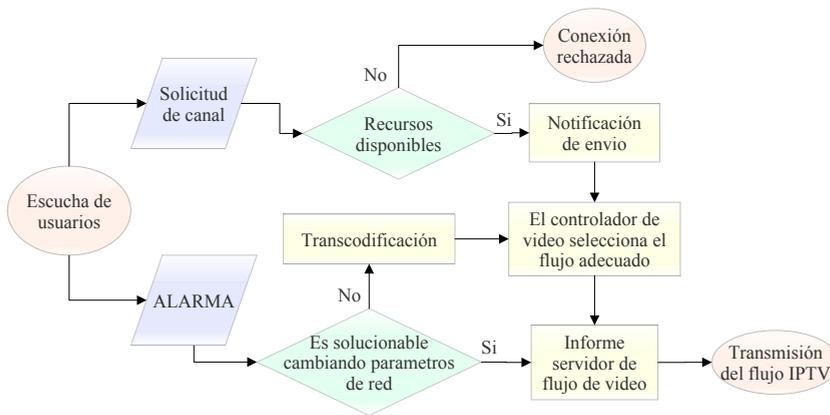


Figura 3.7: Algoritmo simplificado del sistema global de gestión.

En la figura 3.8 y figura 3.9 podemos ver los protocolos básicos de comunicación llevados a cabo cuando se produce una petición de servicio IPTV o cuando se produce una alarma.

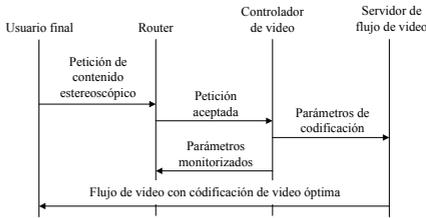


Figura 3.8: Protocolo para la petición de un servicio de IPTV.

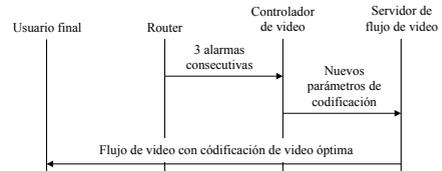


Figura 3.9: Protocolo para la activación de una alarma en el controlador de vídeo.

3.8 Proceso general de análisis de la QoE

Como hemos visto en el apartado anterior, existe un proceso, que hemos denominado *proceso QoE*, encargado de activar la alarma en el caso de que exista una degradación de la QoE. Este proceso está diseñado para operar entre la red del ISP y la red del usuario ya que recopila información de ambas arquitecturas tal como se puede apreciar en la figura 3.10.

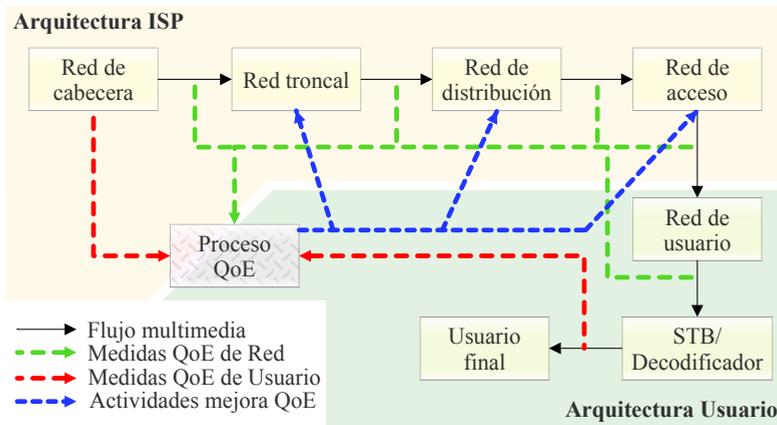


Figura 3.10: Proceso de QoE para el control de la degradación de la QoE.

En el *proceso QoE* continuamente se recaba información de la red y del usuario para obtener parámetros como: el jitter, el retardo, los paquetes perdidos, el tiempo

de zapping, etc... Este proceso se lleva a cabo en el controlador de vídeo. Con los datos recabados de la parte del usuario y con los datos de red se estima un valor de QoE utilizando técnicas de inteligencia artificial. Si este valor es insuficiente para garantizar la QoE durante la entrega del servicio de IPTV, el controlador de vídeo llevará a cabo mejoras de red tales como: dar más ancho de banda a la VLAN, cambio de red de acceso y un conjunto de mejoras más que se pueden consultar en la sección 3.10. Si con estas mejoras no es suficiente para solucionar el problema se realizará transcodificación.

Para llevar a cabo la acción de recabar la información de la red y de los usuarios se dispone de un servidor capaz de enviar traps SNMP a los dispositivos de red correspondientes. Los traps SNMP recibidos nos darán información del estado de la red de distribución, del consumo de los recursos de red y en general de los parámetros necesarios para la estimación de la QoE.

3.9 Banco de pruebas para el estudio de la QoE

La arquitectura básica del banco de pruebas que hemos utilizado para los diferentes estudios de este trabajo de investigación es la que se muestra en la figura 3.11. Como vemos se compone básicamente de un servidor de flujo de vídeo, una red de distribución y un cliente.

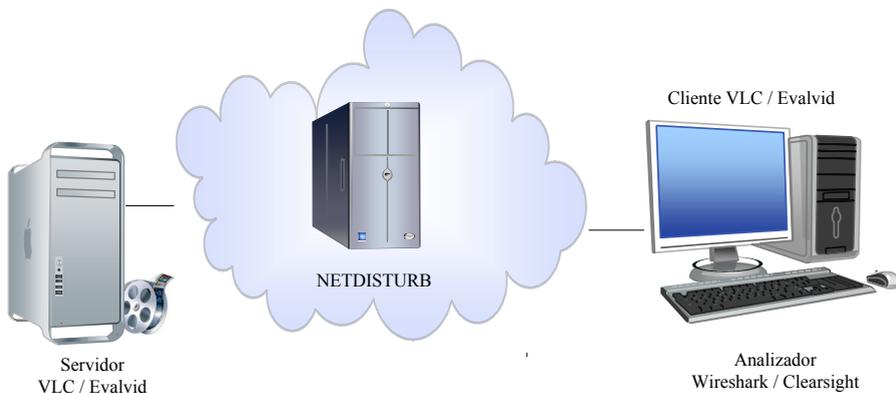


Figura 3.11: Arquitectura del banco de pruebas.

En el siguiente apartado veremos las distintas configuraciones que hemos utilizado para simular y evaluar una transmisión de flujo de vídeo IPTV en base a la arquitectura que acabamos de plantear.

3.9.1 Configuración para la transmisión de vídeo utilizando VLC/Evalvid

3.9.1.1 Evaluación en base a VideoLAN media player (VLC)

Para el servidor de vídeo por lo general se ha utilizado el programa *VLC*. Para su configuración como servidor se selecciona en primer lugar la opción convertir/emitir del menú archivo. Desde la pantalla emergente se configuran las características de la transmisión. Estas características son básicamente: selección del flujo de vídeo, configuración del perfil y configuración del destino. En primer lugar se carga la *ubicación del flujo vídeo* que se va transmitir. A continuación se *configura el perfil* del flujo de vídeo llevando a cabo un proceso de transcodificación. Para ello se selecciona básicamente el formato del contenedor y la codificación del audio y vídeo. Por lo general, la codificación del flujo de vídeo la hemos realizado con el programa *FFMPEG*. Uno de los contenedores más comunes para el transporte del flujo de vídeo, y que más hemos utilizado, es MPEG-TS. El estándar de compresión ya utilizado, dependerá de las características de la transmisión y el destino. Otro parámetro interesante que se podría configurar es el bitrate. Por último se *configura los parámetros de transmisión*. Entre ellos se encuentra: el tipo de protocolo utilizado (HTTP, UDP, RTP ...), el tiempo de vida (Time To Live (TTL)), el puerto y la dirección de destino. Ésta puede ser unicast, multicast o broadcast, dependiendo de cómo deseamos que se envíe el flujo. Si deseamos que se envíe a un cliente en particular se utilizará una dirección unicast. Si queremos que se envíe a un grupo de clientes, realizando así una multidifusión, se utilizará una dirección multicast (con direcciones desde 224.0.0.0 a la 239.255.255.255). Pero si lo que se desea es enviar a todos los clientes, utilizaremos una dirección broadcast del tipo 255.255.255.255. Para evitar la configuración manual mediante el interfaz *VLC*, se podría llevar a cabo la configuración de la transmisión directamente por línea de comandos.

Por ejemplo:

```
> vlc -vvv BBB_16_9_720p_QS_4.mp4 --sout udp:232.32.32.32 --ttl 12
```

Tras la configuración ya se puede iniciar la transmisión de vídeo. El siguiente paso sería configurar el cliente para poder recibir el vídeo y de esta forma poder comprobar la calidad del vídeo recibido. Para configurar el cliente *VLC*, abrimos el programa y seleccionamos abrir red desde el menú de archivo. Tras esta selección se desplegará una ventana en la que tendremos que añadir la Uniform Resource Locator (URL) del servidor a la que nos queremos conectar, indicando tanto el

protocolo utilizado como la dirección de red. Para evitar la configuración manual mediante el interfaz VLC, se podría llevar a cabo la recepción del flujo de vídeo directamente por línea de comandos.

Por ejemplo:

```
> vlc -vvv udp://@232.32.32.32
```

Para analizar los parámetros de red durante la transmisión hemos utilizado el programa *Wireshark*. *Wireshark* es un sniffer utilizado principalmente para el análisis de protocolos, lo que nos permite ver y analizar todo el tráfico que pasa a través de una red. Por tanto, con la información extraída de este programa podemos medir los parámetros de red durante la transmisión. Para analizar la calidad del vídeo recibido hemos utilizado otro tipo de programas que a partir del vídeo recibido degradado y el origen sin degradar, se extraen valores como la VQM. El programa utilizado es *Analyzer ClearSight*.

3.9.1.2 Evaluación en base a Video Quality Evaluation Tool (*EvalVid*)

Hay trabajos en los que en lugar de utilizar el programa *VLC* hemos utilizado el programa *Evalvid*. *Evalvid* nos permite evaluar, entre otros, los parámetros de red (como el jitter, retardo y paquetes perdidos), el MOS, el PSNR y el MIV de una transmisión de flujo de vídeo cliente/servidor. La forma de utilizar esta herramienta de trabajo se explica en (Klaue, Rathke y Wolisz 2003). En esta página web también hay varios tutoriales muy interesantes. Para entender un poco mejor qué es *Evalvid* diremos que: *Evalvid* se trata de un conjunto de herramientas de código abierto (publicadas sin licencia) para la evaluación de la calidad de vídeo transmitido a través de una red de comunicaciones real o simulada. Esta herramienta ha sido desarrollada por el grupo de investigadores *Telecommunication Networks Group*. *EvalVid* está escrito en C. Desde la página web de los autores de *Evalvid* se puede descargar, tanto el código fuente como los binarios compilados para Windows, Linux y MacOSX. Esta herramienta la hemos utilizado sobre todo en aquellos trabajos en los que se ha analizado el PSNR y el MIV. A continuación explicaremos cómo hemos utilizado esta herramienta.

Tras codificar el vídeo con *FFMPEG* se utiliza una de las herramientas de *Evalvid* para encapsularlo, *MP4Box*. Para la evaluación off-line de la calidad de vídeo una vez transmitido se necesita extraer unas características previamente (antes de la transmisión) mediante esta misma herramienta.

Por ejemplo:

```
> MP4Box -hint -mtu 1024 -fps 30 -add a01.m4v a01.mp4  
> MP4Box -std -sdp a01.mp4 > a01.sdp
```

A continuación se crea un vídeo de referencia a partir del vídeo encapsulado utilizando el programa *FFMPEG*. Este vídeo se utilizará para extraer todos aquellos parámetros relacionados con la calidad de vídeo recibido como el PSNR, MIV, etc.

Por ejemplo:

```
> ffmpeg -i a01.mp4 a01_ref.yuv
```

Ahora ya estamos preparados para la transmisión del flujo de vídeo. El programa utilizado para llevar a cabo la transmisión es *mp4trace*. Para su configuración indicaremos la dirección de difusión, el puerto y el flujo de vídeo a transmitir. En ese mismo número de puerto, tanto en el servidor como en el cliente, pondremos a la escucha un programa sniffer similar al *Wireshark* pero sin interfaz gráfica como es el *tcpdump*. Los datos extraídos del sniffer serán utilizados para el análisis de la transmisión.

Por ejemplo:

```
> mp4trace -f -s 192.168.0.2 12346 a01.mp4 > st_a01
> tcpdump -n -tt -v udp port 12346 > sd_a01
> tcpdump -n -tt -v udp port 12346 > rd_a01
```

Los ficheros obtenidos durante el proceso de transmisión son:

<code>a01_ref.yuv</code>	El fichero de vídeo en crudo, antes de ser codificado.
<code>a01.mp4</code>	El fichero de vídeo codificado y encapsulado para la transmisión.
<code>sd_a01</code>	Los paquetes IP enviados desde el servidor.
<code>rd_a01</code>	Los paquetes IP recibidos desde el cliente.
<code>st_a01</code>	Información del emisor como el tipo de trama, de segmentación de paquete, etc.

Para finalizar, a partir de estos ficheros y mediante los programas *etmp4* y *psnr* llevaremos a cabo el proceso de evaluación. A partir de los resultados obtenidos podremos extraer el MIV y el MOS de la siguiente forma.

Por ejemplo:

```
> psnr 352 288 420 akiyo_cif.yuv a01_ref.yuv > ref_psnr.txt
> etmp4 -F -x sd_a01 rd_a01 st_a01 a01.mp4 a01e
> mos /work ref_psnr.txt 25 > mos.txt
> miv /work > miv.txt
```

3.9.2 Configuración de la red de distribución

3.9.2.1 Configuración básica

Para emular las condiciones de la red durante la transmisión del flujo de vídeo hemos utilizado el programa *NetDisturb*. Este programa se instala en un equipo con dos tarjetas de red. El cliente y el servidor irán conectados a cada una de estas tarjetas. Este programa tiene un manual que podemos descargar de www.zti-telecom.com. El programa, por lo general, lo hemos configurado para simular condiciones de red basadas principalmente en:

- Un retardo constante con un jitter exponencial.
- Un retardo constante con un jitter uniforme.
- Pérdida de paquetes basada en %.

Estas son algunas de las configuraciones de red con las que hemos trabajado nosotros, pero se podría configurar cualquier otra. Para ello se puede consultar el manual.

3.9.2.2 Ampliación de la configuración básica

Una forma de ampliar la configuración básica de la red de distribución es añadiendo dispositivos reales tales como routers y switches que simulen el resto de la arquitectura de red. Un ejemplo sería añadir un switch de capa 3 que emulara el núcleo de red, un ordenador y dos routers que simularan la red de distribución y por último un switch junto a un router inalámbrico para simular la red de acceso. En ambos extremos de esta configuración de red tendríamos el proveedor de servicios de IPTV y el cliente. Esta configuración nos permite modificar otro tipo de parámetros tales como grupos multicast, enrutamiento, VLANs, gestión del tipo de colas, etc.

3.10 Teoría de colas, prioridad de servicios y cambios en la red

Uno de los principales problemas en las redes de datos es la congestión. Hay muchos servicios tales como vídeo, VoIP, P2P, etc, que se utilizan masivamente en la actualidad y que hacen que aumente considerablemente el tráfico. IPTV está siendo integrado por muchos operadores de red, utilizando redes de próxima generación (NGN) con el fin de garantizar adecuadamente el nuevo servicio. Las redes IPTV no son inmunes a la congestión. Por esta razón se hace necesario un control de la

congestión para evitarlo. Cuando el tráfico de la red es alto, los servicios que se ofrecen comienzan a degradarse rápidamente. Para evitar la congestión en la red se utiliza en los routers un sistema de control de congestión que consiste en eliminar paquetes al azar una vez que se ha excedido el límite de tráfico de datos. Este sistema tiene la desventaja de que no diferencia la prioridad y el tipo de tráfico que se descarta. Esta técnica se denomina *Traffic Policing*.

A parte de este método para el control de la congestión, el controlador de vídeo puede realizar otros cambios en la red como:

Cambiar el flujo de tráfico: En este caso, el sistema realiza todas las modificaciones en los dispositivos de red. Por ejemplo, el sistema puede modificar el ancho de banda que se utiliza en las redes VLAN para el servicio de IPTV, cambiar el sistema de gestión de colas, cambiar las prioridades de tráfico, etc.

Itinerancia (cambio de la red de acceso): Este caso ocurre cuando la red que está dando el servicio de IPTV no tiene suficientes recursos disponibles.

Cambio a una línea de copia de seguridad: A veces los proveedores de servicios de Internet disponen de dispositivos de copia de seguridad en la red o se pueden alquilar líneas a otros proveedores de servicios con el fin de tener más ancho de banda en la red de distribución.

Cambio de copia de seguridad o router: Podría haber más conmutadores o routers que proporcionen tolerancia a fallos en la red. En este caso podrían ser activados para proporcionar más ancho de banda durante un periodo de tiempo necesario.

Seleccionar otras rutas de enrutamiento: Jitter, retardo y en general los parámetros de tiempo no suelen incluirse en los algoritmos de enrutamiento con el fin de seleccionar la mejor ruta para entregar el contenido de multidifusión. Cuando un camino está sobrecargado, el sistema puede indicar a los routers del núcleo qué camino debe seguir el contenido con el fin de proporcionar un menor jitter, retardo y tiempo de ida y vuelta. Hay que reconocer que este proceso es complejo en servicios multicast, pero existe una solución basada en la división por grupos de los clientes tal como se ha analizado en la subsección 4.4.1.2.

3.11 Conclusiones

En este capítulo hemos definido términos y conceptos importantes que utilizaremos durante el resto del trabajo. Hemos definido qué es la QoS y la QoE. En un entorno de IPTV, definimos la QoE como las sensaciones, percepciones y opiniones que tienen los usuarios al recibir dicho servicio y entendemos por QoS los requisitos necesarios para garantizar su entrega. Para garantizar la QoE en IPTV uno de los aspectos indispensables es la VQ recibida por el usuario. Para medir esta calidad disponemos de tres métodos: los subjetivos, los objetivos y los híbridos. Los subjetivos se basan en la opinión, los objetivos en cálculos matemáticos y los híbridos en la combinación de ambos. La métrica subjetiva utilizada para medir la QoE es el MOS. Una de las métricas objetivas que hemos utilizado mucho en este trabajo para medir la calidad de vídeo es la VQM.

Hemos analizado en este capítulo los parámetros utilizados para medir tanto la QoE como la QoS en entornos de IPTV. Es un aspecto importante ya que en este trabajo vamos a desarrollar un método inductivo de estimación de la QoE y utilizaremos parámetros de QoS en el desarrollo de este método. En concreto estos parámetros son: el jitter, el retardo, los paquetes perdidos y el ancho de banda. El hecho de utilizar estos parámetros es porque en nuestro planteamiento partimos de que existe una correlación entre la QoS y la QoE. Luego, a partir de parámetros de QoS estimamos la QoE. Para analizar la QoE hemos desarrollado un banco de pruebas que utilizaremos en los distintos estudios realizados en este trabajo.

Otro de los aspectos importantes que hemos tratado en este capítulo ha sido la descripción global del sistema de gestión para IPTV. Hemos definido un infraestructura general de red para IPTV en la que está integrado el sistema de gestión. Para explicar el funcionamiento del sistema de gestión hemos desarrollado un algoritmo en el que se detalla cada uno de los procesos que realiza. La mayoría de estos procesos han sido tratados en profundidad en otros capítulos de este trabajo. Tal como se describe en el algoritmo, el sistema de gestión se basa fundamentalmente en dos eventos para su activación: uno es el de alarma y el otro el de solicitud de servicio. Para que se active una alarma debe ocurrir que la estimación de QoE supere un umbral, tal como se ha explicado en la subsección 3.7.2. Para que se active la solicitud de servicio debe ocurrir que un cliente solicite al servidor el servicio de IPTV.

El trabajo realizado ha sido revisado y publicado en diferentes actas de congresos y revistas internacionales. En este capítulo hemos hecho referencia a algunas de estas publicaciones. Por ejemplo, hemos referenciado (Bri y col. 2013) ya que en este artículo se explica como obtener las características de un dispositivo. Esta información es necesaria sobre todo para la gestión de dispositivos móviles tal como se muestra en la descripción del algoritmo de gestión global. Por otro lado, en el estudio realizado en (Canovas y col. 2010b) determinamos los umbrales de

los parámetros de QoS en base a la QoE y cuyos resultados han sido referenciados en este capítulo. Un parámetro de QoS analizado en profundidad en este capítulo ha sido el tiempo de zapping, mostrando por ejemplo el tiempo de zapping para diferentes tecnologías. Esta información ha sido extraída de (García y col. 2009) y (Lloret y col. 2011b). Por último, dentro del estudio realizado en (Lloret y col. 2011b) analizamos en profundidad la VQM. En él desarrollamos una forma de correlacionar la VQM con el MOS y que ha sido presentada en este capítulo.

Capítulo 4

Medidas preliminares

En este capítulo se van a llevar a cabo una serie de medidas preliminares con el fin de analizar procesos que se van a realizar durante la ejecución del sistema de gestión. En la sección 4.1 se va a analizar el proceso de handover y roaming durante la entrega de un servicio de IPTV. Analizaremos cómo afecta este proceso a la QoE y cómo se puede gestionar. En la sección 4.2 analizaremos los diferentes formatos de codificación y contenedores para vídeo estándar, HD, 3D/ST y dispositivos móviles. En la sección 4.3 analizaremos en profundidad el proceso de codificación y en la sección 4.4 el de transcodificación.

4.1 Roaming y Handover

Para ofrecer movilidad a un usuario durante una transmisión de flujo de vídeo, son necesarias soluciones que permitan la interoperabilidad entre los proveedores y las tecnologías inalámbricas. Por tanto, son necesarios modelos de itinerancia y handover que permitan la interconexión entre diferentes tecnologías de acceso inalámbricas de banda ancha. Por definición, roaming es la capacidad de seguimiento de la posición de los dispositivos móviles. El roaming incluye tanto la actualización de la información de ubicación de un usuario móvil (usuario roaming), como el reencaminamiento del tráfico de usuario entrante a la nueva ubicación de red del usuario (Meyer 2006). Mientras que handover es el proceso por el cual se transfiere una comunicación de un canal de una célula a un nuevo canal, en la misma célula o en otra. Según cómo se realice dicha transferencia podemos diferenciar entre:

Handover seguro (Soft-Handover): éste se produce cuando se mantiene el dispositivo móvil conectado a dos canales simultáneamente y hasta que la recepción del nuevo no mejora la anterior, no se produce este traspaso.

Handover forzado (Hard-Handover): el salto de un canal a otro se produce en un momento determinado sin que exista periodo de transición.

Si el proceso anterior se produce dentro de una misma tecnología, se considera como *handover horizontal* (intra-tecnología). En cambio, si el paso de una red a otra es entre diferentes tecnologías se considera que se está llevando a cabo un *handover vertical* (inter-tecnologías). Un proceso de handover en general, permite al usuario pasar de una celda a otra sin pérdidas e incluso sin interrupciones de los servicios que se están utilizando en ese momento. Los principales factores para llevar a cabo el proceso de handover según (Yan, Ahmet Şekercioğlu y Narayanan 2010) son: la potencia de la señal recibida, el ancho de banda disponible, el consumo de energía, el coste, la calidad del servicio, la interferencia, la seguridad, las preferencias del usuario y el handover forzado.

Como hemos dicho, la operación de handover permite el mantenimiento de las conexiones en curso mientras un usuario se mueve a través de diferentes redes de acceso inalámbrico. Para ello, el proceso de handover requiere redireccionar el tráfico entrante y saliente de una conexión en curso con el fin de no causar ninguna interrupción desde el punto de vista del usuario. Es por esto que uno de los parámetros más importantes durante este proceso es el tiempo necesario para llevar a cabo el handover. Algunos estudios, como el presentado en (Zaidi y col. 2011), tratan de disminuir este parámetro mediante el uso de técnicas de posicionamiento. En este apartado analizaremos cómo influye el proceso de handover en la QoS de una transmisión de flujo de vídeo. Analizaremos también cómo se lleva a cabo este proceso basándonos en la potencia de la señal recibida o RSSI. Este proceso se utiliza en una de las fases del sistema de gestión, concretamente durante la gestión de usuarios conectados al servicio de IPTV a través de una red inalámbrica. Este algoritmo de gestión se presenta en la subsección 6.2.2.

4.1.1 Proceso de Handover en base a la RSSI

El problema de la QoE en servicios de IPTV se complica en entornos inalámbricos por dos motivos principalmente: por la limitación del ancho de banda y por el proceso de handover. En este subapartado explicaremos, en base a (Canovas y col. 2012), cómo se lleva a cabo el proceso de conexión/desconexión en base a la RSSI durante la entrega de un servicio de IPTV con el fin de incluirlo en el sistema de gestión. En la figura 4.1 podemos ver el diagrama de flujo de dicho proceso.

Tal como vemos en la figura 4.1, el proceso de handover desde el punto de vista del cliente móvil durante un servicio IPTV consta de 3 fases principalmente:

- Petición de asociación a los Access Point (AP).
- Evaluación y selección del AP óptimo.

■ Conexión multicast.

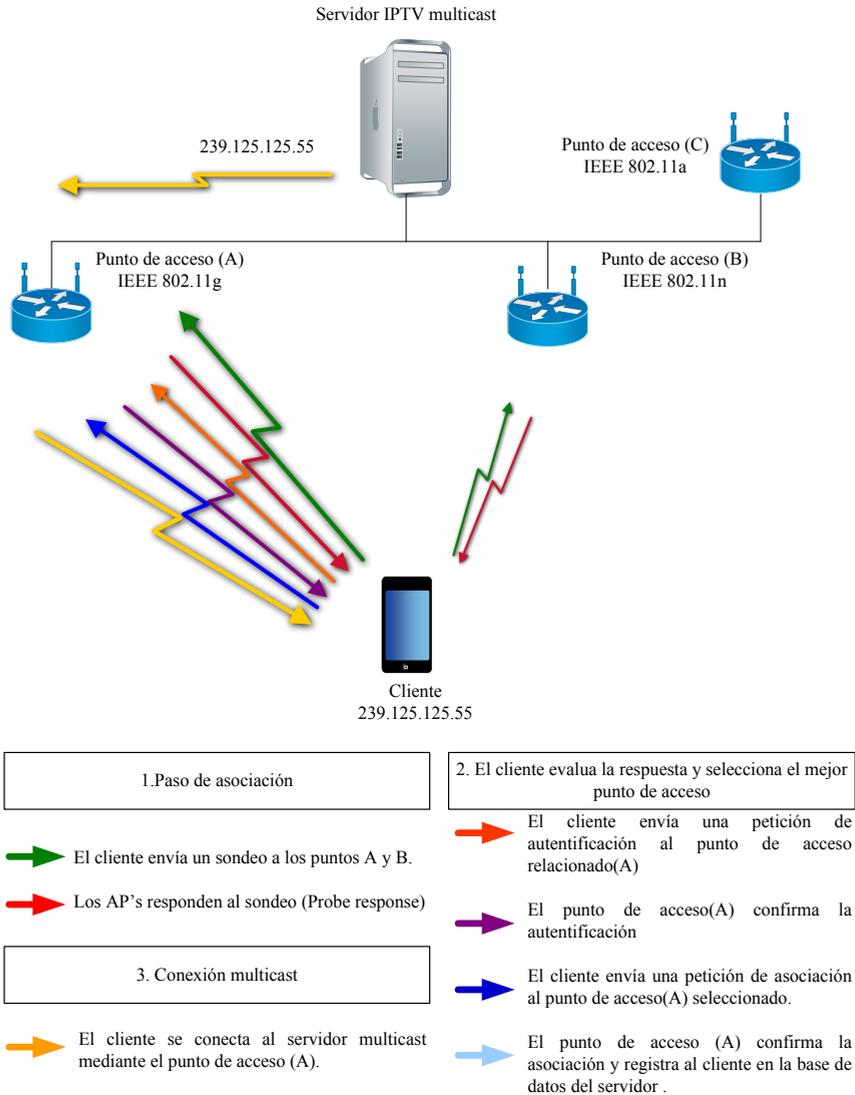


Figura 4.1: Proceso de handover de un cliente IPTV en un entorno inalámbrico.

En primer lugar el cliente sondea los APs más próximos, es decir, aquellos con mayor RSSI. Los APs responden a esta petición. Esta información se envía al servidor de test de QoE. Tras su evaluación envía al cliente a qué AP debe conectarse. A continuación, el cliente inicia el proceso de autenticación y posterior asociación con el AP seleccionado. Al finalizar este proceso el cliente ya puede iniciar o reanudar la conexión multicast de IPTV a través del AP seleccionado. Para llevar a cabo este proceso desarrollamos un protocolo de acceso inalámbrico representado en la figura 4.2.

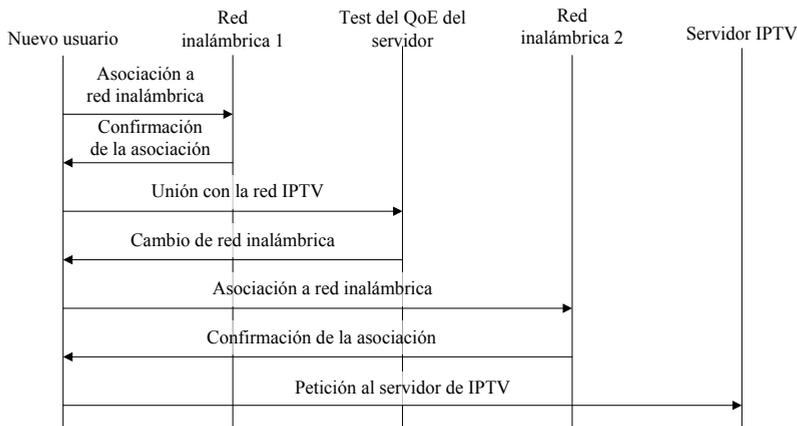


Figura 4.2: Protocolo de acceso a un servicio de IPTV en redes inalámbricas

Bajo el punto de vista del servidor, cuando el controlador de vídeo recibe un mensaje de una red IPTV con información para estimar la QoE de esa red, la compara con la información de redes inalámbricas dentro del área de cobertura de esa misma red. Si el servidor detecta una red inalámbrica dentro de esa área de cobertura con la que se pueda mejorar la QoE, enviará un mensaje al cliente de cambio de red inalámbrica. En este mensaje se envía información de cuál es la red a la que debe conectarse. Tras la reconexión, el cliente solicita el flujo IPTV al servidor a la nueva ubicación, con un mensaje de petición.

Como se ha podido observar, este sistema conecta al cliente con bajo QoE o a los nuevos clientes a aquellas redes inalámbricas con las que se mejora la QoE (si las hay), evitando así que todos se conecten al mismo canal o red. Las redes inalámbricas con un número alto de usuarios conectados normalmente experimentan mayores valores de retardos, jitters y pérdidas de paquetes con la consecuente

degradación de la QoE. En el siguiente apartado analizaremos cómo influye el proceso de handover en dichos parámetros de red.

4.1.2 Caso de estudio: análisis del proceso de handover durante una transmisión de vídeo

4.1.2.1 Escenario

En primer lugar describiremos el escenario utilizado para llevar a cabo este caso de estudio. Como podemos ver en la figura 4.3, la red se divide en dos partes: una parte cableada correspondiente a la red de distribución del servidor IPTV y otra inalámbrica correspondiente a la red de acceso del cliente IPTV. Cada uno de los clientes utiliza una tecnología IEEE 802.11a/g/n diferente. Con el fin de evitar cualquier dependencia con el fabricante, utilizamos un punto de acceso que fuese capaz de ofrecer cada una de las tecnologías inalámbricas. Esta topología es ideal para este caso de estudio, ya que, aunque es muy simple, ofrece la posibilidad de analizar el rendimiento de la red en cada uno de los casos de handover planteado. Por otra parte, nos permite analizar diferentes condiciones de entorno en el proceso de transmisión de vídeo. En la siguiente subsección describiremos las medidas realizadas en base a este escenario y comentaremos los resultados obtenidos.

4.1.2.2 Medidas Realizadas

El proceso llevado a cabo para la toma de medidas fue el siguiente. Iniciamos el proceso transmitiendo un flujo de vídeo desde el servidor IPTV al cliente y nos aseguramos de que la recepción es correcta. El cliente recibe el flujo de vídeo a través de una tecnología inalámbrica concreta. Dado cierto tiempo forzamos el dispositivo a cambiar de tecnología. Las medidas realizadas se pueden observar en la tabla 4.1.

Durante todo el proceso analizamos los parámetros de red mediante un sniffer (Wireshark) con el fin de comprobar cómo influye la itinerancia de una tecnología a otra en la QoS. Los parámetros que analizaremos son: el jitter, el retardo, los paquetes perdidos y los paquetes por segundo. Para analizar el proceso de handover con más detalle mostraremos únicamente 100 muestras antes y 100 después de que se haya producido.

En la figura 4.4 se muestra el número de paquetes recibidos por segundo. Hemos tomado 20 segundos antes y 20 segundos después de la itinerancia. Observamos que el valor mínimo de paquetes recibidos es entre 95 y 128 paquetes/seg para todos los casos. Sin embargo, cuando se analiza el valor medio de los paquetes/seg recibidos, es fácil ver que la itinerancia de IEEE 802.11n a IEEE802.11a y desde IEEE802.11n a IEEE 802.11g proporciona valores mucho más altos que en el resto

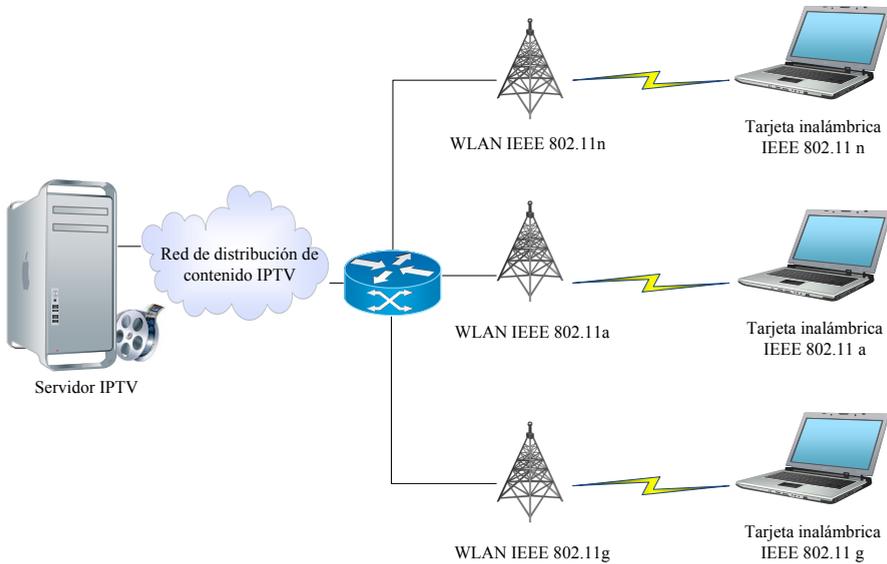


Figura 4.3: Proceso de handover de un cliente IPTV en un entorno inalámbrico.

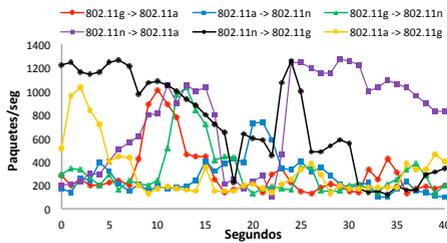
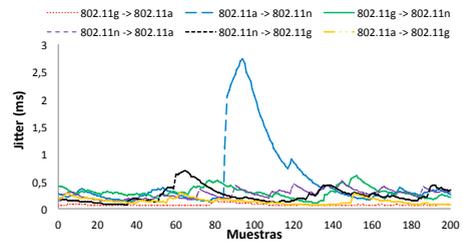
de casos (alrededor de 750 paquetes/seg). De hecho, si comparamos este valor con la itinerancia de IEEE 802.11a a IEEE802.11n se reduce en casi un tercio (alrededor de 270 paquetes/seg). Los casos restantes presentan un valor medio en torno a 305 paquetes/seg.

En la figura 4.5 se muestra el jitter durante el proceso de handover entre diferentes tecnologías inalámbricas. En este caso, es fácil ver que la itinerancia que más afecta al jitter es cuando se pasa de IEEE 802.11a a IEEE802.11n, con un pico de 2,73ms. El valor medio durante el resto de la transmisión es de 0,53ms. El resto de casos presentan un valor medio entre 0,25ms y 0,30ms, con la excepción de la itinerancia de IEEE 802.11a a IEEE802.11g y de IEEE 802.11g a IEEE802.11a, que presentan valores entre (0,139ms y 0.074ms. respectivamente).

En la figura 4.6 se muestra el retardo durante el proceso de handover entre diferentes tecnologías inalámbricas. Si analizamos con detalle cada medida, podemos observar que aquellas transiciones cuya tecnología origen es IEEE 802.11a presenta valores de retardo mucho más altos que el resto. Estos valores se encuentran entre

Tabla 4.1: Handover entre diferentes tecnologías.

Tecnología Origen	Frecuencia de trabajo	Tecnología destino	Frecuencia de trabajo
IEEE 802.11a	5 GHz.	IEEE 802.11g	2,4 GHz.
IEEE 802.11a	5 GHz.	IEEE 802.11n	2,4 GHz.
IEEE 802.11n	2,4 GHz.	IEEE 802.11a	5 GHz.
IEEE 802.11n	2,4 GHz.	IEEE 802.11g	2,4 GHz.
IEEE 802.11g	2,4 GHz.	IEEE 802.11a	5 GHz.
IEEE 802.11g	2,4 GHz.	IEEE 802.11n	2,4 GHz.

**Figura 4.4:** Número de paquetes recibidos por segundo.**Figura 4.5:** Jitter por muestra.

5ms y 6ms, mostrando picos superiores a 20ms. El resto de casos tienen un valor promedio entre 2ms y 4ms, con valores de pico inferiores a 10ms en todos los casos.

La figura 4.7 muestra el número de paquetes perdidos por muestra durante el proceso de handover entre diferentes tecnologías inalámbricas. Por un lado, se observa que para todos los casos se produce un pico de pérdida de paquetes alrededor de la muestra 100 (el punto donde se realiza la itinerancia entre tecnologías). Los casos que presentan mayor número de paquetes perdidos son cuando se cambia de IEEE 802.11a a IEEE 802.11g y de IEEE 802.11g a IEEE802.11a. En estos casos se producen unas pérdidas de alrededor del 18% por cada muestra. Todos los demás casos tienen valores entre el 8% y el 13%.

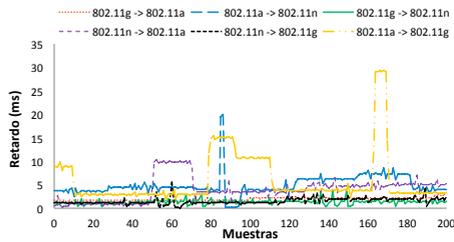


Figura 4.6: Retardo por muestra.

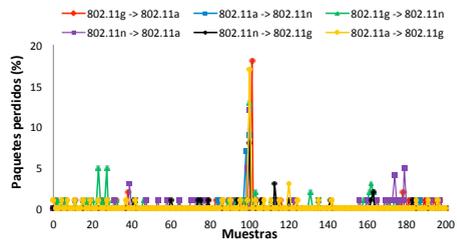


Figura 4.7: Número de paquetes perdidos en % por muestra.

4.2 Compresión y contenedores

Los sistemas de compresión/descompresión y procesamiento de señales digitales (Digital Signal Processor (DSP)) son comúnmente utilizados en las comunicaciones de voz y vídeo con el fin de disminuir ancho de banda. Esta disminución debe cumplir una compensación entre calidad y ancho de banda. Luego, con la compresión se proporciona más ancho de banda pero siempre buscando obtener la menor degradación de la calidad posible. La efectividad de una técnica de compresión de imágenes viene dada por la relación de compresión. Se calcula como el cociente entre el tamaño del archivo de la imagen original (sin comprimir) y el tamaño del archivo de imagen resultante (comprimida). A mayor relación de compresión menor ancho de banda utilizado y menor calidad de imagen resultante.

Las técnicas de compresión pueden ser aplicadas a nivel de imagen o a nivel de vídeo. A nivel de imagen se busca las similitudes entre píxeles. A nivel de vídeo se busca las similitudes entre imágenes. Un ejemplo de compresión a nivel de imagen es Joint Photographic Experts Group (JPEG) y a nivel de vídeo es MPEG. Para llevar a cabo la extracción de estas similitudes se utilizan parámetros del vídeo como: el espacio, el tiempo, la psicovisualidad y la propia codificación del vídeo. La redundancia espacial se da cuando entre dos píxeles contiguos existe muy poca varianza o tienen un alto grado de correlación. La redundancia temporal se da cuando entre dos imágenes seguidas existen mínimas diferencias. La redundancia psicovisual se da cuando las diferencias visuales son imperceptibles para el ojo humano. Por último, la redundancia en la codificación se da cuando en una imagen ciertos parámetros de similitud tienen la misma probabilidad de que se den. Para este caso se utiliza el mismo número de bits de codificación.

Un CÓDEC es un CODificador-DECodificador, es decir, un programa que es capaz de comprimir y codificar vídeo/audio en un formato determinado y de decodificarlo y descomprimirlo, para posteriormente poder ser visto. Los códecs se utilizan para digitalizar y comprimir cualquier vídeo analógico. La elección del códec tiene mucha importancia ya que determinará la calidad y la tasa de bits del vídeo final.

En una transmisión de vídeo, la codificación influye en la robustez ante pérdidas de datos, errores de transmisión, retrasos en la transmisión, etc... La característica más importante que debe cumplir la codificación de vídeo en una transmisión es la escalabilidad. A partir de un único flujo de transmisión cada usuario debe disponer de la mejor calidad posible. Para ello se debe tener en cuenta principalmente parámetros como: la resolución de pantalla, el bitrate o bits por segundo, las imágenes por segundo y relación señal a ruido. Todos estos parámetros estarán limitados principalmente por el ancho de banda del canal.

4.2.1 Estándares de compresión: familia MPEG y serie ITU H-XXX

El estudio y desarrollo de nuevos estándares de compresión ha sido un proceso continuo en las últimas décadas. La finalidad de estas investigaciones se han centrado principalmente en las necesidades de velocidad tanto de transmisión como de reproducción de aplicaciones de vídeo. La publicación de estos estándares se ha llevado a cabo como recomendaciones en la *Unión Internacional de Telecomunicaciones* (UIT), de la serie H, así como diferentes formatos definidos por el subgrupo de trabajo ISO/IEC JTC 1/SC 29/WG 11 perteneciente al *Moving Picture Expert Group* (MPEG), comité del organismo *International Organization Standardization* (ISO) e *International Electrotechnical Commission* (IEC), responsable de los estándares de la familia MPEG: (entre los que destacamos MPEG-1, MPEG-2 y MPEG-4). En las tabla 4.2 y tabla 4.3 clasificaremos algunos de los estándares de compresión más conocidos según: familia MPEG y serie ITU H-XXX.

Tabla 4.2: Tabla de códecs de vídeo Familia MPEG.

Códec	Características	Descripción
MPEG-1	Formato utilizado en Compact Disc (CD) y Video CD (VCD). Comprime a 1.5 Mbps.	Se basa en el uso de la Discrete Cosine Transform (DCT) y en la eliminación tanto de la redundancia como de la entropía.
MPEG-2	Formato utilizado en Digital Television (DTV) y High Definition Television (HDTV). Opera en un rango de 2 a 15 Mbps, aunque puede funcionar a velocidades superiores de 100Mbps.	Utiliza la DCT y codificación entrópica. Tiene un amplio rango de aplicaciones, velocidades, resoluciones, calidades y servicios.

Continúa en la página siguiente. . .

Tabla 4.2 – Continúa

Códec	Características	Descripción
MPEG-4 Parte 2/H.263	Se utiliza en entornos donde se requieren baja tasa de bits y una baja resolución tales como transmisiones de vídeo.	Es un estándar de compresión basado en la DCT, similar a estándares anteriores como MPEG-1 y MPEG-2.
DivX	Formato de compresión basado en los estándares MPEG-4.	Utiliza la DCT. Permite elegir el número de fotogramas I, P y B, las velocidades de codificación, los intervalos máximos de fotogramas clave, la compensación de movimiento, el tipo de cuantificación y el método entrelazado.
XviD	Formato de compresión basado en el estándar MPEG-4 ASP.	Permite elegir el número de imágenes I, P y B, la compensación de movimiento global, la cuantización, el tipo de cuantificador, el ocultamiento de luminancia y el entrelazado.
H. 264/MPEG-4 AVC	Realiza una compresión de vídeo de alta calidad a una tasa de bits sustancialmente más bajas que los estándares MPEG-2, H.263 y MPEG-4 Parte 2. Además de las imágenes I, P y B introduce dos nuevas, la SP (Switching P) y la SI (Switching I) que sirven para codificar la transición entre dos flujos de vídeo.	En un principio se enfocó para aplicaciones de videoconferencia de baja calidad y aplicaciones de Internet. Posteriormente se amplió para: soportar resoluciones más elevadas, un muestreo de 4:4:4 o 4:2:2, poder mezclar escenas, tasas binarias más elevadas, poder representar algunas partes de vídeo sin pérdidas y utilizar el sistema de color por componentes RGB.
Ogg Theora	Se utiliza con bitrates bajos. Soporta datos de vídeo progresivos. Es compatible con los formatos de submuestreo de croma: 4:2:0, 4:2:2 y 4:4:4. Únicamente soporta intra-imágenes (I) e inter-imágenes (P).	Es un método de compresión de vídeo con pérdidas. El vídeo comprimido puede ser almacenado en cualquier contenedor multimedia, pero generalmente se encuentra en el formato contenedor .ogg que es el recomendado por Xiph.Org. Este códec utiliza la DCT 8x8 de Tipo-II y compensación de movimiento basada en bloque.

Tabla 4.3: Tabla de códecs de vídeo serie ITU H-XXX.

Códec	Características	Descripción
H261	Sistema de codificación con pérdidas.	Códec para compresión de vídeo digital. Se utiliza para servicios de videotelefonía y videoconferencia a 64 kbps. Calidad inferior a MPEG.
H263	Compatible con MPEG-4 parte 2. Su compresión es mayor que H261.	Se utiliza en videotelefonía. Utiliza la DCT. Realiza una compresión temporal intensa. Funciona preferiblemente con secuencias de vídeos que presentan cambios reducidos entre imágenes.
H264	Alcanza mayor calidad de imagen que MPEG-2 con la mitad de tasa binaria. Permite distribuir con una calidad de vídeo excelente a través de todo el espectro posible comprendido entre 3G y la HD (desde 40 Kbps hasta más de 10 Mbps).	Desarrollado por el equipo Joint Video Team (JVT). Utiliza la DCT, un algoritmo híbrido de predicción y transformación para la reducción de la correlación espacial y de la señal residual, control de la tasa binaria, predicción por compensación de movimiento para reducir la redundancia temporal y la codificación entrópica para reducir la correlación estadística.
M-JPEG	Utiliza la tecnología de codificación intra-frame, la cual es muy similar a la parte I-frame de los estándares de codificación MPEG-1 y MPEG-2 pero sin emplear la predicción inter-frame. El formato tiene una tasa de bits relativamente alta dada la calidad entregada.	Se utiliza frecuentemente en sistemas de edición de vídeo no lineal donde cada cuadro o frame de una secuencia de vídeo digital es comprimido por separado como una imagen JPEG. MJPEG es considerado actualmente un códec desactualizado e ineficiente.

Continúa en la página siguiente. . .

Tabla 4.3 – Continúa

Códec	Características	Descripción
Windows Media Vídeo	WMV es un formato de compresión de vídeo que comprende un conjunto de algoritmos de compresión desarrollados por Microsoft. Tiene una tasa binaria alta.	Soporta bitrate variable, medio y constante. A partir de la versión 9 se introdujeron nuevas características como soporte nativo para vídeo entrelazado, píxeles no cuadrados y la interpolación de cuadros. WMV es ideal para VoD .
Real Vídeo	Su capacidad de transmisión lo hace ideal para transmitir en tiempo real en red. Alta tasa de compresión.	Soportado por múltiples plataformas, incluyendo Windows, Mac, Linux, Solaris y varios terminales de telefonía móvil.
AVI	Formato definido por Microsoft e IBM para el intercambio de datos multimedia en plataforma Windows.	Existen dos tipos de codificaciones AVI: las basadas en vídeos para Windows (VfW) y las basadas en DirectShow.

4.2.1.1 Caso de estudio: análisis del ancho de banda necesario para diferentes códecs de compresión

El banco de pruebas para llevar a cabo este estudio fue el siguiente. Simulamos una red de IPTV con una arquitectura básica según planteamos en la sección 3.9. En uno de los extremos de la red conectamos el servidor de vídeo IPTV y en el otro el usuario final. Tanto para el cliente como para el servidor utilizamos el programa VLC. Para la captura de los paquetes y medir la calidad de vídeo recibido se utilizó el programa ClearSight Analyzer. Las características del vídeo con el que se realizaron las pruebas son: codificación MPEG-2, resolución 720x576, 25 fps, calidad Digital Versatile Disc (DVD) estándar y una duración de 120 minutos.

Una vez descrito el banco de pruebas analizamos cuál es el ancho de banda que se necesita dependiendo del formato de compresión que se utilice, con el objetivo de obtener un modelo de anchos de banda de referencia. Para comparar la capacidad de compresión de los diferentes códecs realizamos las siguientes tareas. En primer lugar seleccionamos una tasa de bits fija para que este parámetro no influyera en el ancho de banda. El valor seleccionado fue 2000Kbps ya que es el valor máximo común para todos los códecs que nos permite el programa que utilizamos (*Total*

Video converter). Los formatos de compresión que analizamos fueron: MPEG-2, MPEG-4, H.264, Divx y Xvid, siendo MPEG-2 el códec de referencia para el estudio. El motivo de la selección de estos formatos de codificación es porque son de los más utilizados. Para la prueba se transmitió el vídeo codificado en cada uno de los formatos propuestos y medimos el ancho de banda recibido en la parte del cliente. Los resultados obtenidos se muestran en la figura 4.8.

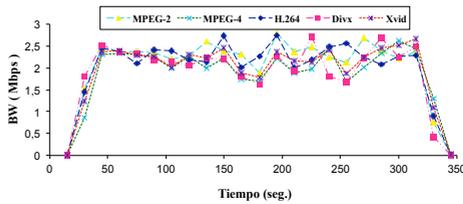


Figura 4.8: Ancho de banda utilizado por diferentes códecs durante la transmisión de un flujo de vídeo.

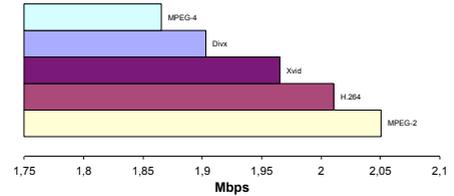


Figura 4.9: Ancho de banda medio utilizado por cada uno de los códecs.

Tal como vemos en la figura 4.8 la transmisión se inicia en el segundo 15 y tiene una duración de aproximadamente 350s. También se pueden observar valores de pico que representan ráfagas de tráfico. Esto es debido a que al comprimir se obtienen diferentes valores de tasa de bits dependiendo sobre todo de si la imagen es del tipo I, P o B. Para comparar mejor la capacidad de compresión de cada uno de los formatos podemos observar la figura 4.9, en la que se representa el valor medio del ancho de banda utilizado por cada uno de los formatos de codificación durante la transmisión. Claramente MPEG4 es el formato que más comprime. Comparando cada uno de los formatos con respecto a MPEG-2 se obtiene como resultado los valores de la tabla 4.4. Vemos que MPEG-4 reduce hasta un 9,26% la tasa de bits.

En la tabla 4.4 se puede observar la importancia de los sistemas de codificación a nivel de reducción de la tasa de bits y la capacidad de compresión de alguno de los códecs más utilizados. Estos resultados se han extraído de uno de nuestros estudios, en concreto de (Lloret y col. 2011b).

Para ampliar cualquier información con respecto a este caso de estudio se puede consultar (Atenas y col. 2010).

Tabla 4.4: Tabla comparativa del porcentaje de compresión de diferentes códecs respecto a MPEG-2

Formato	Compresión (%)
H.264	1.95
Xvid	4,39
Divx	7.31
MPEG-4	9.26

4.2.2 Formatos contenedores multimedia

Un contenedor es un formato de archivo multimedia en el que los datos y metadatos se almacenan en formato de meta-archivo sin ser codificados. Por tanto, podemos decir que los vídeos son encapsulados en contenedores para su transporte manteniendo toda su información del tipo: el flujo de vídeo y audio, los metadatos, los subtítulos, el capítulo de información, la información de sincronización, etc. A continuación mostraremos el formato de contenedores más típicos y sus características principales.

<code>.wmv</code>	La extensión de archivo <code>.wmv</code> suele describir los archivos ASF que utilizan códecs de WMV.
<code>.avi</code>	WMV se pueden almacenar en un archivo AVI cuando se utiliza el compresor de vídeo WMV-9. AVI deriva del formato de intercambio de archivos de recursos (RIFF) el cual divide los datos del archivo en bloques Es uno de los más utilizados.
<code>.mkv</code>	Es compatible con los siguientes códecs de compresión: WMV, DivX, MPEG y MPEG4 La combinación de un vídeo comprimido en MP4 y encapsulado en mkv es muy común. Soporta códecs 3D.
<code>.mp4</code>	Este formato contenedor se corresponde con MPEG-4 parte 14. Permite flujos a través de redes IP aunque es más común para almacenar vídeo digital y audio Soporta códec 3D.
<code>.3gp</code>	Se utiliza generalmente en teléfonos móviles/PDAs para servicios multimedia de 3G/4G y UMTS. Almacena vídeo en formato MPEG-4, H.263 y H.264.
<code>.3g2</code>	Es muy similar a 3gp pero con la diferencia de que 3g2 fue diseñado para teléfonos basados CDMA y 3gp para GSM.
<code>.mov</code>	Formato estándar nativo del programa QuickTime, desarrollado por Apple que también utiliza tecnología compatible con MPEG-4 y posee unos códecs muy avanzados que permiten resultados muy superiores a otros estándares.
<code>.mpg</code>	Soporta normalmente las codificaciones MPEG-1/2.
<code>.ogg</code>	Se utiliza típicamente para almacenar datos en archivos para ordenadores y para transmitir flujos audiovisuales. Contiene normalmente DivX, Xvid, Theora, Vorbis, Speex o FLAC.
<code>.webm</code>	Para transmisiones de archivos sobre una conexión HTTP. Normalmente encapsula VP8.
<code>.rmvb</code>	El contenedor real media con bitrate variable se usa habitualmente con RealVideo. Se obtienen archivos más pequeños que con formatos como MPEG.
<code>.flv</code>	Utilizado para transmitir vídeo por internet utilizando flash.
<code>.vob</code>	No soporta códecs 3D pero sí MPEG-1/2.

4.2.3 Códecs de compresión y contenedores para móviles

En la tabla 4.5 se muestra los códecs y contenedores más utilizados en dispositivos móviles.

Para saber si el sistema de gestión es capaz de transcodificar en tiempo real formatos de codificación para dispositivos móviles realizamos el siguiente caso de estudio.

Tabla 4.5: Características de vídeo de diferentes dispositivos móviles.

Mobile	Vídeo					
	Contenedor	Códec	Resolución	Rel. Aspecto	fps	Bitrate (Kbps)
Nokia NEC Samsung Siemens,..	.3gp	H.263	176:144	11:9	12.5	144
Motorola Sony	.3g2	MPEG4 part12	176:144	5:4	25	144
iPhone	.mp4	MPEG4 part12	640:480	4:3	25	765
iPod	.mp4	MPEG4 part12	320:240	4:3	23.976	720
Android	.mp4	MPEG4 part12	480:320	3:2	15	1200
Black Berry	.avi	H.263	176:144	11:9	25	300
Pocket PC (Medium quality)	.wmv	WMV8	320:240	4:3	25	768
Pocket PC (Low quality)	.wmv	WMV8	176:144	5:4	25	300
Others	.wmv	WMV8	320:176	16:9	25	768
Others	.ogg	Ogg Theora	320:240	4:3	25	384

4.2.3.1 Caso de estudio: análisis de los formatos de compresión de cada dispositivo móvil

En este caso de estudio vamos a analizar ciertos aspectos relacionados con los dispositivos móviles y sus formatos de codificación como: el consumo de ancho de banda, la tasa de compresión y el tiempo de transcodificación. Para ello, seleccionamos un vídeo en crudo con las siguientes características: una duración de 600 seg, una resolución 1280x720 pixels, una relación de aspecto de 16:9, un bitra-

te de 4456Kbps a 23.976 fps y con tamaño total de 346.5 Mbytes. Para calcular los tiempos hemos utilizado un PC con 2 Gigabyte (GB) de Random-Access Memory (RAM) y un procesador Intel Pentium 4 a 3.0 Gigahertz (GHz), utilizando el software *FFMPEG*.

En el primer experimento que realizamos, analizamos el tiempo necesario por cada uno de los dispositivos móviles en realizar un proceso de transcodificación. El objetivo de este experimento es conocer si el servidor de vídeo es capaz de realizar transcodificación en tiempo real. Para ello, se ha calculado el tiempo necesario en comprimir el vídeo en crudo en cada uno de los formatos de codificación móvil. En la figura 4.10 podemos observar los resultados obtenidos. El menor tiempo utilizado es para los sistemas con contenedores 3GP, 3G2 y para los dispositivos Pocket PC de baja calidad. El mayor tiempo es para los dispositivos iPhone y para aquellos formatos de codificación de vídeo con contenedores OggTheora. Realizamos otro experimento muy relacionado con éste en el que comprobamos el tiempo que necesita el servidor para comprimir únicamente un segundo del vídeo. Los resultados obtenidos se muestran en la figura 4.11. Los que menos tiempo necesitan son los formatos de codificación de vídeo con contenedores 3GP y con una codificación de H.263. Los que más tiempo necesitan son los dispositivos iPhone y aquellos formatos de codificación de vídeo con contenedores OggTheora.

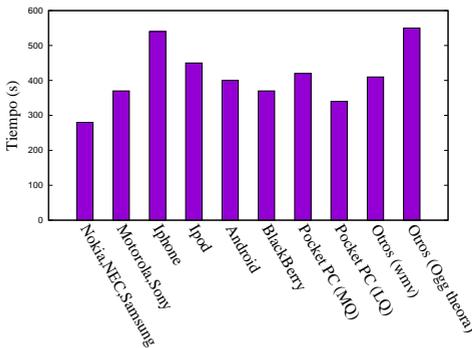


Figura 4.10: Comparación del tiempo de transcodificación.

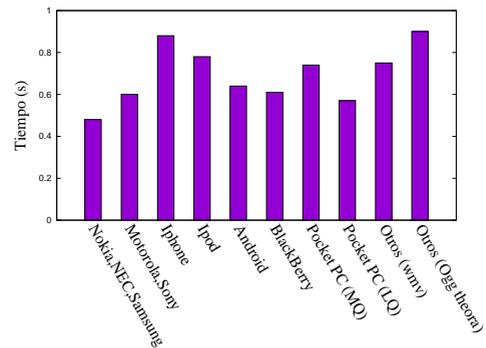


Figura 4.11: Tiempo para comprimir un segundo de vídeo.

Con el fin de conocer el consumo de ancho de banda según el tipo de móvil reproducimos en local cada vídeo en cada uno de los formatos correspondiente a cada dispositivo móvil durante los 600 segundos que dura el vídeo. Los resultados obtenidos en este experimento se pueden observar en la figura 4.12. Como vemos los que menos ancho de banda consumen son los que utilizan contenedores 3GP y 3G2. También son los que tienen una tasa de vídeo menor, inferior a 144Kbps. Los que más ancho de banda consumen son los dispositivos Android, HTC Hero y iPhone.

Para comparar la capacidad de compresión por cada uno de los dispositivos móviles se llevó a cabo una comparación con respecto a la tasa de compresión. Según podemos observar en la figura 4.13 los que mayor tasa de compresión han obtenido son aquellos que utilizan contenedores 3GP y 3G2. Los que peor resultado han dado son Android y HTC Hero seguidos de iPhone y Pocket PC con calidad media. Hemos observado que todos ellos tienen una resolución de 176:144.

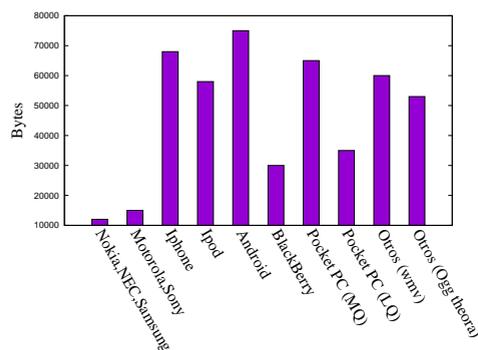


Figura 4.12: Comparación del ancho de banda consumido.

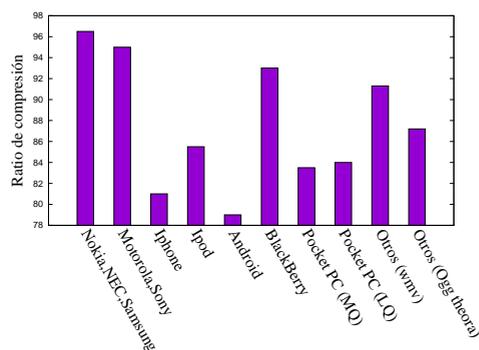


Figura 4.13: Comparación de la tasa de compresión.

4.2.4 Códecs de compresión 3D/ST

Tabla 4.6: Tabla de códecs de vídeo 3D/ST

Códec	Características	Descripción
Multi-View Video Coding (MVC)	Utiliza una alta tasa de bits ya que transmite a la vez dos o más imágenes.	Basado en la visión estereoscópica desde varios puntos de vista.
Multi-View Video Depth (MVD)	Reduce el bitrate.	Basado en la información estereoscópica en niveles de negro o sombras. Consiste en a una imagen 2D más otra con la información de profundidad en niveles de grises 2D+Z.

En este apartado describiremos los principales conceptos de los formatos de codificación 3D. Los formatos que vamos a tratar son: codificación 2D+Z y codificación de vídeo estereoscópica.

2D+Z (Oh y col. 2007) es un formato de codificación 3D. Utiliza dos imágenes: una para la componente de color y otra para el mapa de profundidad. La imagen del mapa de profundidad está en escala de grises. Para cada pixel de la imagen de color se asigna un valor de profundidad y se genera la imagen 3D. La figura 4.14 muestra un ejemplo del formato 2D+Z. La principal característica de esta codificación es que reduce el ancho de banda utilizado con respecto a la codificación estéreo. Esto ocurre porque sólo utiliza una imagen con toda la información y otra con la información de profundidad en escala de grises. El rango de profundidad está limitado por la distancia máxima y mínima a la cámara. Para comprimir la secuencia de vídeo con formato 2D+Z se ha utilizado el estándar de compresión H.264 Auxiliary Picture Syntax (Wiegand y Sullivan 2007) con un códec H.264/AVC. La secuencia de vídeo 3D resultante se puede ver en TV especiales como 3DWOW de (*Philips 3D Solutions*).



Figura 4.14: Imagen extraída de un vídeo con formato 2D+Z.

El formato de codificación de **vídeo estereoscópica** se compone de dos secuencias de vídeo cada una tomada desde un punto de vista diferente. Una de las secuencias corresponde con la visión izquierda y la otra con la derecha. Podríamos decir que es exactamente igual que lo que hace la visión humana. Al ver a través de ambos ojos se genera una vista del entorno desde el ojo izquierdo y otra desde el derecho. A continuación es nuestro cerebro el encargado de crear las imágenes 3D. Las técnicas de codificación de vídeo basadas en este formato toman en cuenta la similaridad entre la imagen derecha y la izquierda con el fin de reducir la cantidad de datos. El hecho de utilizar dos imágenes para obtener la imagen 3D implica que necesitamos el doble de ancho de banda para el transporte de este tipo de vídeo a través de la red. Un ejemplo del formato de codificación de vídeo estéreo se muestra en la figura 4.15.

Para comprimir la secuencia de vídeo con formato estéreo se ha utilizado el estándar H.264/AVC Multi-View Video Coding el cual es una extensión del estándar H.264 Auxiliary Picture Syntax (Vetro, Wiegand y Sullivan 2011). Codifica utili-



Figura 4.15: Imagen extraída de un vídeo con formato estéreo.

zando técnicas de compresión temporal inter-cuadro y predicción entre vistas. Este estándar tiene diferentes perfiles dependiendo del formato de codificación que se utilice. En concreto, el perfil de alto estéreo (Stereo High Profile) está diseñado específicamente para vídeo estereoscópico de 2 vistas. En el caso de que se quisiera aumentar el número de vistas se utilizaría un perfil alto multivista (Multiview High Profile).

La forma de compresión de ambos formatos es la siguiente. Para el formato de vídeo estéreo, la secuencia de vídeo estereoscópica cruda, sin comprimir, se creó uniendo la secuencia de la vista izquierda con la de la derecha. Para codificar y decodificar ambos vídeos, se utilizó el programa Joint Multiview Coding Video (*JMVC*). El proceso de codificación para el formato 2D+Z se llevó a cabo de la misma forma que el anterior. La diferencia está en que la secuencia de vídeo sin comprimir se creó a partir de una de las vistas más la información de profundidad respecto a la vista en cuestión. En este sentido, el segundo caso tiene más complejidad que el primero ya que se debe estimar la información de profundidad utilizando el algoritmo más adecuado.

En uno de los estudios que realizamos, en concreto en (Lloret y col. 2011a), comparamos la capacidad de compresión entre los formatos 2D+Z y estereoscópica. El resultado obtenido se muestra en la tabla 4.7.

Como se puede observar en la tabla 4.7 la capacidad de compresión del formato 2D+Z es mucho mayor que la del vídeo estéreo, prácticamente el 50%. Esto es debido a que la mitad de la información de las secuencias 2D+Z se corresponde con información de profundidad y cuyos valores están en escala de grises. Las imágenes en escala de grises a la hora de ser comprimidas tienen una alta redundancia tanto espacial como temporal.

Tabla 4.7: Comparación códecs estereoscópicos

Códec	Bitrate (Kbps)
Estéreo	6.440.311
2D+Z	3.428.352

4.3 Cuantificación

La cuantificación es un procedimiento llevado a cabo por un codificador y consiste en reducir la precisión de los coeficientes de la DCT con el fin de obtener una tasa de bits más baja. En la reconstrucción, el decodificador realiza el proceso contrario al realizado en la cuantificación, de tal forma que la matriz de coeficientes $Q_F(v, u)$ es inversamente cuantificada dando como resultado los coeficientes de la DCT reconstruidos $F(v, u)$. La expresión matricial de coeficientes para MPEG2 resultante se obtiene mediante la siguiente ecuación extraída de la recomendación (ITU-T/H.262 2012):

$$F(v, u) = \sum_{v=0}^7 \sum_{u=0}^7 (2^n Q_F(v, u) k) a + \left[\left(2Q_F(v, u) + \left(\frac{d|Q_F(v, u)|}{d(v, u)} (1 - k) \right) \right) W(w, v, u) \frac{Q_s}{32} \right] (1 - a) \quad (4.1)$$

Dónde,

- 2^n representa el multiplicador intra DC (intra-bloque del codificador) para $n \in 0, 1, 2, 3$. En el caso de MPEG-2, se estima de acuerdo con la tabla 7-4 de la recomendación (ITU-T/H.262 2012).
- El valor de la variable a depende de los intra-bloques. Si se cumple que $v = 0$, $u = 0$ y $k = 1$ entonces $a = 1$. $a = 0$ para el resto de casos.
- La segunda parte de la expresión depende del valor de k , la luminancia y la crominancia. $k=1$ en el caso de intra-bloques y $k = 0$ si no lo son.
- Q_s corresponde al factor de cuantificación

El Q_s (o QS) es un entero que se codifica con una de longitud fija de 5 bits. Comprende valores dentro del rango 1, ..., 31. El Q_s con un valor igual 0 no está

permitido. Cada coeficiente de $W(w,v,u)$ es un valor entero con un tamaño de 8 bits. Comprenden valores entre $w = 0 \dots 3$, $v = 0 \dots 7$ y $u = 0 \dots 7$.

La variación del Q_s afecta a la tasa de bits de un vídeo o imagen tal como se puede observar en la tabla 4.8, extraída de nuestro trabajo (Canovas y col. 2010b). De tal forma que cuanto mayor sea el valor del Q_s menor será la tasa de bits. Hay que tener en cuenta que una reducción de la tasa de bits conlleva una disminución de la calidad de imagen. Por ello debe existir una compensación entre el valor de Q_s y la calidad de imagen resultante. En un codificador, el Q_s se puede cambiar en el comienzo de la codificación de cada macro bloque. En el caso de IPTV, el Q_s puede ser recuperado de forma muy simple en el bloque del decodificador MPEG de IQ (Cuantificación inversa) de un receptor de IPTV o STB.

Tabla 4.8: Comparación de la capacidad de compresión en tasa de bits (Kbps) para distintos valores de Q_s .

Formato	QS1	QS4	QS6	QS31
MPEG2	37355	19000	12029	2377
MPEG4	31452	13898	8406	1833

A continuación realizaremos dos casos de estudio con el fin de comprobar cómo afecta el Q_s a la VQ y cómo afecta el Q_s a las condiciones de red.

4.3.1 Caso de estudio: cómo afecta el Q_s a la VQ

El objetivo de este caso de estudio es comprobar cómo afecta el Q_s a la VQ. Para ello realizamos el siguiente análisis. En primer lugar codificamos un vídeo de referencia MPEG1 con un formato utilizado en televisión de acuerdo con la definición estándar SDTV a partir de 14315 imágenes Portable Network Graphics (PNG). Lo ajustamos a una resolución de 720x576 dpi a 25 frames por segundo. A continuación, a partir de este vídeo de referencia codificamos varios vídeos con diferentes valores de Q_s (Canovas y col. 2010b) tanto en MPEG-2 como en MPEG-4. El software utilizado para el proceso de codificación fue *FFMPEG*. Los valores de Q_s utilizados fueron (1, 2, 4, 6, 8, 10, 16, 31). A continuación calculamos la VQ comparando el vídeo de referencia con el codificado. Para ello utilizamos el programa *Elecard Video Quality Estimator*. Los resultados obtenidos se muestran en la figura 4.16 y la figura 4.17.

Los mejores resultados obtenidos en cuanto a la de calidad de vídeo tanto para MPEG-2 como para MPEG-4, tal como era de esperar, fueron con $Q_s = 1$ y

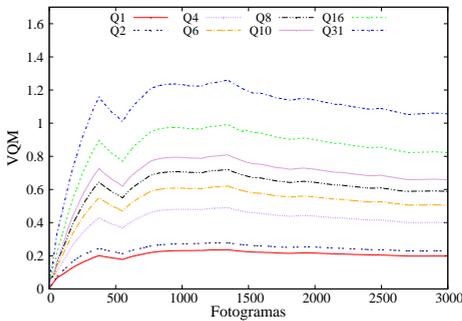


Figura 4.16: Calidad de vídeo objetiva en recepción para diferentes QS en vídeos Mpeg2.

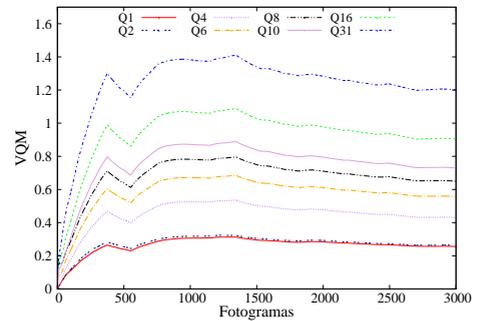


Figura 4.17: Calidad de vídeo objetiva en recepción para diferentes QS en vídeos Mpeg4.

$Q_s = 2$. También, tal como esperábamos, los peores fueron con $Q_s = 16$ y $Q_s = 31$. Con estos dos últimos valores se ha obtenido una calidad muy baja por lo que se descartarían para una transmisión de flujo de vídeo IPTV. Cabe destacar la diferencia de calidad entre la codificación MPEG-2 y la MPEG-4. La calidad obtenida con MPEG-2 es mejor que con MPEG-4 para todos los valores de cuantificación Q_s .

En relación a este caso de estudio y como resultado del trabajo de investigación llevado a cabo en (Lloret y col. 2011b), planteamos un método objetivo basado en RR para la estimación de la VQ según la (4.2).

$$V_q = Q_s + FS + PLR \quad (4.2)$$

Donde, Q_s sería el factor de escala, FS (Frame Statistic) un valor estadístico obtenido a partir de las últimas n imágenes I/P/B y la información de referencia parcial como podría ser la tasa de paquetes perdidos (Packet Loss Rate (PLR)). Esta información parcial del vídeo se podría obtener a través de un canal auxiliar de datos.

En base a los resultados obtenidos en este estudio podemos afirmar que: al aumentar el valor del Q_s disminuye la calidad de vídeo. Por último, es importante que hagamos una apreciación con respecto a este estudio. En el estudio hemos asignado valores de Q_s constantes aunque un codificador de MPEG comercial para la transmisión de flujo de vídeo utilizaría un Q_s variable por cada frame (I, B, P) o por cada macro-bloque. Por lo que en realidad los valores estudiados de Q_s no debemos considerarlos como valores constantes sino como valores medios. De

esta forma al cambiar el bitrate de un vídeo deberíamos configurar el codificador mediante un bitrate medio de un valor Q_s determinado.

4.3.2 Caso de estudio: cómo afecta el Q_s a las condiciones de red

Ampliamos el caso de estudio anterior con el fin de analizar cómo responden los distintos valores de cuantificación a diferentes condiciones de red. Para ello, simulamos una transmisión de IPTV en un entorno real. La arquitectura utilizada para este banco de pruebas es una arquitectura básica basándonos en la descripción de la sección 3.9. Utilizamos el reproductor multimedia *VLC* como servidor de vídeo IPTV multicast. Utilizamos los vídeos del caso de estudio anterior cuyos formatos de codificación eran: MPEG-2 y MPEG-4 con $Q_s = [1, 2, 4, 6, 8, 10, 16, 31]$. Transmitimos cada uno de estos flujos de vídeo en MPEG-2 TS. Instalamos el programa *Wireshark* en el receptor para capturar los paquetes MPEG-2 TS y medir los parámetros de red en recepción. Para visualizar el vídeo recibido utilizamos *VLC* como cliente. Simulamos una red TCP/IP con diferentes parámetros de red uniendo el cliente y el servidor a un mismo PC con el programa *NetDisrupt* instalado. Las diferentes condiciones de red se basaron principalmente en la variación del retardo, el jitter, los paquetes perdidos y el ancho de banda. En la figura 4.18 se puede observar el banco de pruebas que acabamos de plantear.

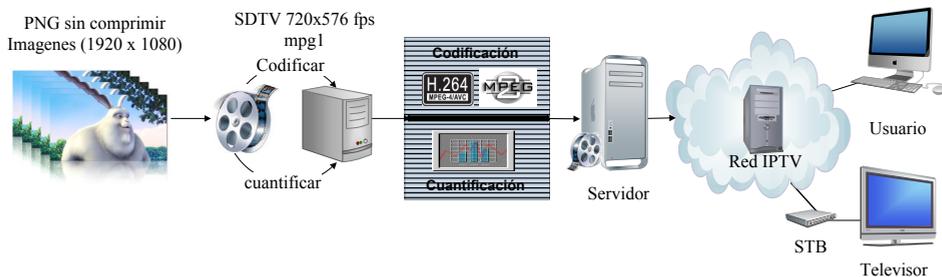


Figura 4.18: Proceso de codificación y ubicación de los dispositivos IPTV

4.3.2.1 Análisis del ancho de banda

Un factor muy importante a tener en cuenta en las transmisiones de flujo de vídeo es el BW. En las figura 4.19 y figura 4.20 se muestra cuál es el ancho de banda necesario en una transmisión para cada uno de los formatos de codificación. Tal como se puede observar el ancho de banda necesario para el $Q_s = 1$ y $Q_s = 2$ es considerablemente superior al resto. En base a este resultado podemos afirmar que: para una transmisión de flujo de vídeo se debe buscar un equilibrio entre calidad necesaria y recursos disponibles.

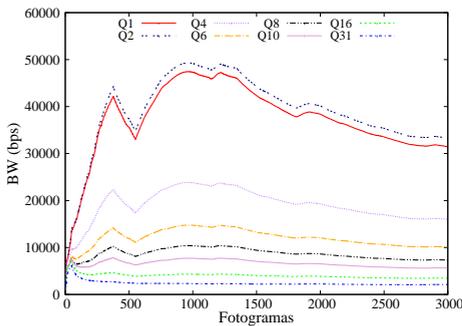


Figura 4.19: Ancho de banda de una transmisión de vídeo Mpeg2 a diferentes QS.

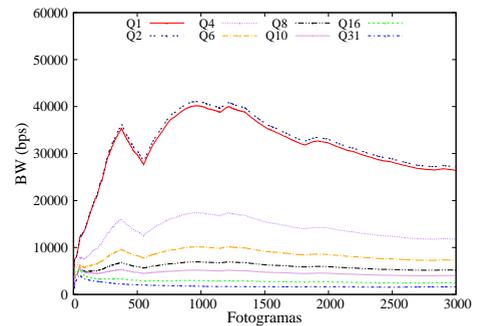


Figura 4.20: Ancho de banda de una transmisión de vídeo Mpeg4 a diferentes QS.

A continuación analizamos cómo responden los diferentes formatos de codificación a distintas condiciones de red. Para ello introdujimos tanto para el análisis del retardo como para el del jitter unas tasas de pérdidas de paquetes del 0.1 %, 1 % y 3 %. El motivo de elegir estas pérdidas es porque estudios como (Rahrer, Fiandra, Wright y col. 2006) perteneciente al forum DSL WT-126, recomiendan un máximo de pérdidas de 5 paquetes IP consecutivos cada 30 minutos en SDTV, o tal como se indica en el trabajo (Canovas y col. 2010b), entre el 0,5 % y 1,5 % de pérdidas de paquetes es aceptable. En base a estos valores estaremos analizando condiciones aceptables de transmisión y condiciones que no lo son y veremos qué repercusiones tienen. Para el estudio del retardo introdujimos un retardo variable entre 2 y 200ms ya que según (ITU-T/G.114 2003) los límites de retardo recomendables en base a la percepción humana son: Bueno entre 0 y 150 ms, aceptable entre 150 y 300ms y pobre mayor que 300ms. Por último, para el estudio del jitter introdujimos un jitter exponencial con valores entre 20 y 50ms ya que según (Calyam y col. 2004) los límites de jitter recomendables en base a la percepción humana son: bueno entre 0 y 20ms, aceptable entre 20 y 50ms y pobre mayor que 50ms. Tanto el retardo como el jitter que introducimos comprendían valores que no dejaban de ser aceptables, ya que nuestro verdadero propósito es ver cómo influyen las pérdidas en la red. El

motivo es que como ya hemos visto en la subsección 6.1.2 el jitter y el retardo, al contrario que las pérdidas de paquetes, son factores que no influyen de forma muy directa en la calidad de vídeo recibida. Además, aplicando técnicas de buffering (e.g., mediante el uso de jitter/playout buffer), estos valores de jitter se pueden compensar, aislando así el impacto negativo de este parámetro sobre la QoE.

4.3.2.2 Análisis del Q_s en condiciones de retardo variable y pérdidas de paquetes en la red

En este análisis comprobamos el retardo con diferentes formatos de codificación y distintas tasas de pérdidas de paquetes en la red.

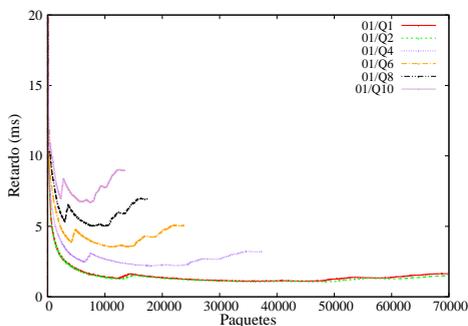


Figura 4.21: Retardo cuando el ratio de pérdidas de paquetes es del 0.1 % para diferentes Q_s en vídeos Mpeg2.

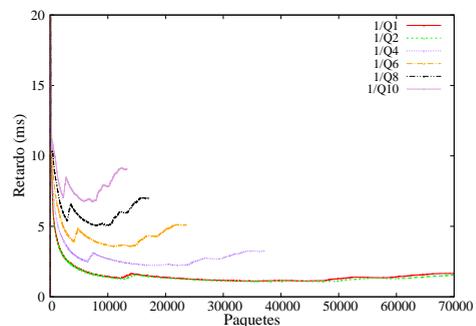


Figura 4.22: Retardo cuando el ratio de pérdidas de paquetes es del 1 % para diferentes Q_s en vídeos Mpeg2.

En la figura 4.21 se puede observar el retardo medio cuando hay pérdidas en la red del 0.1 % durante la transmisión de un flujo de vídeo MPEG2 a diferentes valores de cuantificación. Se puede observar que al aumentar el Q_s aumenta el retardo medio en cada instante de tiempo. También se puede comprobar que a menor Q_s el número de paquetes transmitidos es menor. Por ejemplo, para un Q_s de 6, el retardo es inferior a 5 milisegundos y los paquetes transmitidos superan los 24000. Sin embargo, para un Q_s de 4, el retardo es inferior a 3 milisegundos y los paquetes transmitidos superan los 38000. En la figura 4.22 las pérdidas se aumentaron al 1 %. En cuanto a los resultados obtenidos con respecto al retardo fueron muy similares al caso anterior. Sin embargo, al aumentar las pérdidas al 3 % se puede observar, figura 4.23 , que existe un aumento del retardo durante la transmisión comparado con los dos casos anteriores.

El siguiente experimento fue comprobar las diferencias entre una transmisión de flujo de vídeo MPEG-4 y MPEG-2. Para ello realizamos las mismas pruebas que

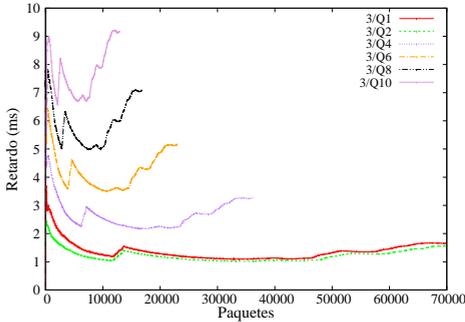


Figura 4.23: Retardo cuando el ratio de pérdidas de paquetes es del 3 % para diferentes QS en videos Mpeg2.

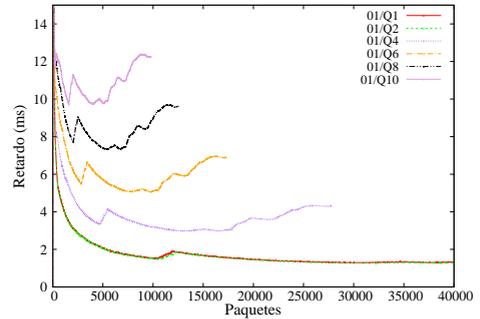


Figura 4.24: Retardo cuando el ratio de pérdidas de paquetes es del 0.1 % para diferentes QS en videos Mpeg4.

para MPEG-2 pero utilizando la codificación MPEG-4. Los resultados obtenidos se pueden observar en la figura 4.24, en la figura 4.25 y en la figura 4.26.

Según los resultados obtenidos el comportamiento entre ambas codificaciones es muy similar, las diferencias son meramente cuantitativas. Destacar que el número de paquetes transmitidos para MPEG-4 es inferior al de MPEG-2 para todos los casos como era de prever, ya que la capacidad de compresión de MPEG-4 es superior al de MPEG-2.

4.3.2.3 Análisis del Q_s en condiciones de jitter y pérdidas de paquetes en la red

En el siguiente estudio se comprobó cómo afecta el Q_s al jitter en una transmisión de flujo de vídeo con pérdidas en la red. Dado que el jitter provocado en la red es del tipo exponencial los resultados obtenidos siguen este comportamiento salvo cuando las pérdidas de paquetes son considerables. Podemos observar como en la figura 4.27 y en la figura 4.28 el comportamiento del jitter es exponencial. Sin embargo, en la figura 4.29 al aumentar las pérdidas de paquetes a un 3 % el comportamiento cambia. Esto es debido a que cuando hay una pérdida de paquetes bastante alta en la red el jitter ya no sigue una distribución concreta. Lo mismo ocurre para MPEG4.

Tal como se puede comprobar en la figura 4.27 y en la figura 4.29 el jitter es menor con un $Q_s=10$. Sin embargo, cuando las pérdidas son del 1 % (figura 4.28) da un jitter menor con $Q_s=1$. Por tanto, el Q_s es un factor a tener en cuenta en una transmisión de IPTV con jitter en la red.

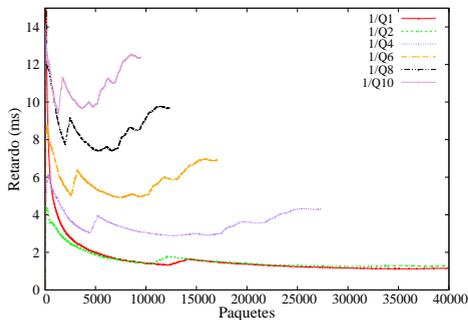


Figura 4.25: Retardo cuando el ratio de pérdidas de paquetes es del 1 % para diferentes QS en videos Mpeg4.

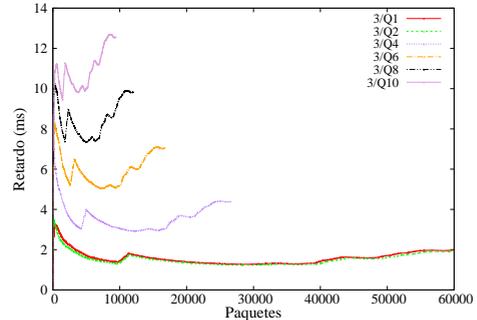


Figura 4.26: Retardo cuando el ratio de pérdidas de paquetes es del 3 % para diferentes QS en videos Mpeg4.

En MPEG4 al haber unas pérdidas del 0.1 % en la red, $Q_s=1$ y $Q_s=6$ son los que mayor jitter provocan durante la transmisión y $Q_s=10$ y $Q_s=2$ el que menos. Con 1 % de pérdidas $Q_s=1$ es el que más jitter provoca. Con el 3 % de pérdidas $Q_s=8$ es el que menos jitter provoca y $Q_s=1$ junto con $Q_s=2$ los que más. En cuanto al patrón de comportamiento ocurre lo mismo que con MPEG2, con 0.1 % y 1 % de pérdidas es exponencial y con el 3 % no, tal como podemos comprobar en la figura 4.30, la figura 4.31 y la figura 4.32.

Para ampliar la información con respecto a este caso de estudio se puede consultar (Canovas y col. 2010b).

4.4 Transcodificación

Un transcodificador de vídeo es un sistema que convierte una secuencia de vídeo en otra con diferentes características (Fernández-Escribano y col. 2006). Las razones por las que se lleva a cabo una transcodificación pueden ser: mejorar el rendimiento, mejorar la calidad de imagen, aumentar el ancho de banda disponible, etc... Como podemos ver en la figura 4.33, los principales parámetros de transcodificación son: la tasa de bits, los frames o imágenes por segundo (frames per second (fps)), la resolución espacial, la sintaxis de codificación y el contenido .

Según (Bjork y Christopoulos 1998), una de las funciones de la transcodificación es dar escalabilidad a los dispositivos multimedia que utilicen un servicio de IPTV. Para ello, el servicio debe ajustarse tanto al dispositivo como al entorno de red utilizado. Un transcodificador de vídeo puede convertir una alta tasa de bits de un programa de TV (como HDTV), en un programa de televisión de más baja

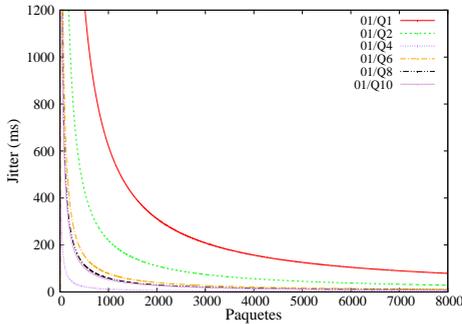


Figura 4.27: Jitter cuando el ratio de pérdidas de paquetes es del 0.1% para diferentes QS en videos Mpeg2.

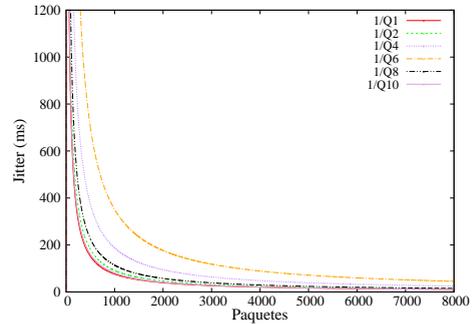


Figura 4.28: Jitter cuando el ratio de pérdidas de paquetes es del 1% para diferentes QS en videos Mpeg2.

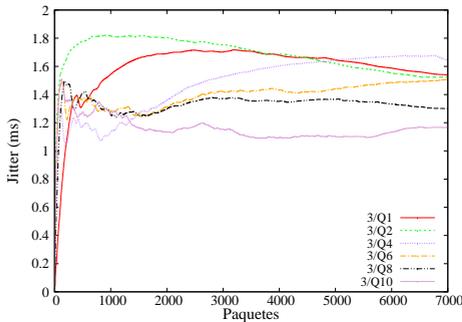


Figura 4.29: Jitter cuando el ratio de pérdidas de paquetes es del 3% para diferentes QS en videos Mpeg2.

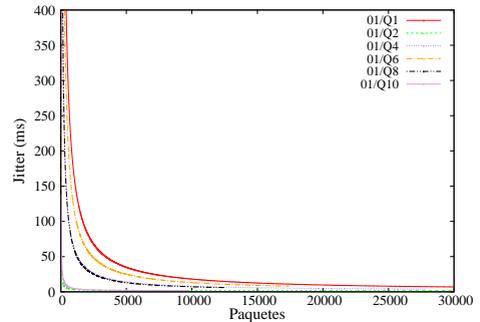


Figura 4.30: Jitter cuando el ratio de pérdidas de paquetes es del 0.1% para diferentes QS en videos Mpeg4.

tasa de bits (como SDTV), con el objetivo por ejemplo de transmitir más canales en la misma red IP. Pero esta reducción de la tasa de bits puede provocar una disminución en la calidad del vídeo transmitido. Por tanto, un transcodificador de vídeo, al igual que un codificador, debe tener un compromiso entre la tasa de bits y la calidad visual.

Un transcodificador en cascada sirve para decodificar completamente un flujo de vídeo y codificar dicho flujo creando una nueva secuencia de vídeo con nuevos parámetros de codificación (Bjork y Christopoulos 1998). Esta concatenación de un decodificador y un codificador, tal como podemos observar en la figura 4.34, le da al transcodificador el atributo de en cascada. El problema del transcodificador

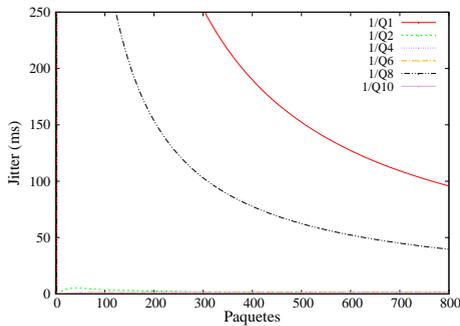


Figura 4.31: Jitter cuando el ratio de pérdidas de paquetes es del 1% para diferentes QS en vídeos Mpeg4.

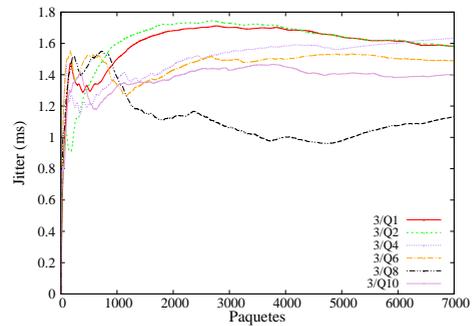


Figura 4.32: Jitter cuando el ratio de pérdidas de paquetes es del 3% para diferentes QS en vídeos Mpeg4.



Figura 4.33: Conversion del formato utilizando un transcodificador de vídeo.

en cascada es que es muy costoso computacionalmente, porque incluye un módulo de estimación de movimiento.

En el transcodificador en cascada de la figura 4.34 se detalla la concatenación entre los procesos de reducción de redundancia temporal y de redundancia espacial. Se inicia el proceso con la reducción de la redundancia temporal, en el que se busca reducir información por predicción de movimiento entre imágenes (ME), proporcionando vectores de movimiento en los píxeles detectados. En la reducción de la redundancia espacial, la imagen diferencia entre la entrante y la precedida se codifica empleando la técnica de la transformada discreta del coseno (DCT) a nivel de bloque. Este algoritmo permite la transformación de áreas de la imagen en representaciones bidimensionales de energía frente a frecuencia. A continuación se aplica un proceso de cuantificación (Q), con pérdidas de información controladas por el nivel de llenado de memoria de salida de los datos codificados y mediante el proceso de codificación entrópica (Variable Length Coding, VLC). La diferencia entre el nivel de llenado y el de vaciado de esta memoria actúa como regulador del valor de cuantificación y, por tanto, sobre el factor de compresión y la calidad

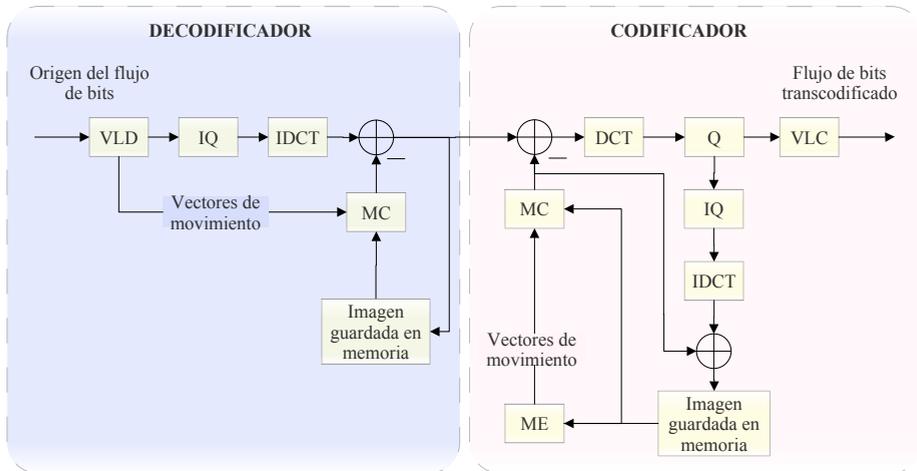


Figura 4.34: Transcodificador en cascada.

final. Este proceso se aplica en menor medida a las zonas de baja frecuencia, en las que el ojo es más sensible. La imagen comprimida resultante es reconstruida mediante los procesos inversos (IQ e IDCT). En el decodificador el procedimiento es el inverso al anterior, con la pequeña diferencia que representa el decodificador de longitud variable (VLD).

Para ampliar cualquier información con respecto a este caso de estudio se puede consultar (Atenas y col. 2010).

4.4.1 Caso de estudio: mejora de un servicio de IPTV con límites de BW utilizando transcodificación

La aplicación de técnicas de transcodificación mejoran el rendimiento de los sistemas de transporte de flujos de vídeo y audio al reducir el ancho de banda necesario para ello. En esta sección lo demostraremos.

Antes de iniciar el estudio es interesante que nombremos los anchos de banda típicos utilizados en diferentes entornos de red ya que nos servirán de referencia en el estudio. Estos valores son: para un canal *RDSI* europeo de banda ancha 2 Megabits per second (Mbps), para un canal típico MPEG-2 estándar (SDTV) 3 Mbps y al rededor de 8 Mbps para cada canal de televisión de alta definición (HDTV), el

ancho de banda típico ofrecido por un proveedor de servicios de IPTV 6 Mbps , para una red inalámbrica *IEEE 802.11b* 11 Mbps y para una red inalámbrica *IEEE 802.11g* 54 Mbps. El número de canales simultáneos dependerá del ancho de banda ofrecido por el proveedor de servicios (2 o 3 canales de *SDTV*, además de 1 canal de *HDTV*). Teniendo en cuenta esta información pasaremos a iniciar el estudio.

4.4.1.1 *Análisis de la transcodificación en transmisiones IPTV*

En este apartado analizamos la importancia de aplicar la transcodificación en transmisiones IPTV. El banco de pruebas para llevar a cabo este estudio fue el siguiente. Simulamos una red de IPTV con una arquitectura básica según planteamos en la sección 3.9. En uno de los extremos de la red conectamos el servidor de vídeo IPTV y en el otro el usuario final. Tanto para el cliente como para el servidor utilizamos el programa VLC. Para la captura de los paquetes y medir la calidad de vídeo recibido se utilizó el programa *Analyzer ClearSight*. Las características del vídeo con el que se realizaron las pruebas son: codificación MPEG-2, resolución 720x576, 25 fps, calidad DVD estándar y una duración de 120 minutos. En la simulación, transmitimos en primer lugar el vídeo con las características que acabamos de nombrar y capturamos el BW recibido en el usuario. Los resultados estadísticos obtenidos se muestran en la tabla 4.9. Cabe destacar el comportamiento tan irregular que tiene la transmisión con respecto al ancho de banda consumido, con picos desde 3.3Mbps hasta 8.5Mbps y una varianza de 2,92Mbps con respecto a la media de 6,2Mbps. Aparecen continuas ráfagas durante la transmisión. Como hemos visto en la introducción de este caso de estudio, una red IPTV transmite más de un canal simultáneamente. Luego si se pretendiera transmitir más de un canal bajo estas condiciones lo más probable es que se diera un caso de congestión en la red produciéndose pérdidas de paquetes, lo que afectaría tanto a la QoS como a la QoE. Una forma de solucionar este problema es realizando transcodificación.

Para llevar a cabo la transcodificación en tiempo real se utilizó el software VLC como transcoder, ya que tiene la capacidad de enviar flujos de vídeo y transcodificar simultáneamente. Repitiendo el experimento anterior pero habiendo realizado transcodificación obtuvimos los resultados que se muestran en la tabla 4.9. Para este caso el ancho de banda utilizado es bastante regular por lo que no aparecen ráfagas durante la transmisión del flujo de vídeo. Estadísticamente, el valor de la varianza respecto al ancho de banda consumido es del 0.38Mbps con respecto a una media de 5.2Mbps. Esta escasa variabilidad del ancho de banda consumido se debe a que utilizamos una tasa de bit constante (CBR).

Si comparamos los resultados obtenidos en la transmisión del flujo de vídeo con y sin transcodificación, vemos que al realizar la transcodificación el ancho de banda consumido disminuye un 16.1% de media. También que la varianza de los datos

Tabla 4.9: Comparativa del proceso de transcodificación

BW (Mbps)	Sin Transcodificación	Con Transcodificación	Reducción (%)
Mínimo	3.3	1.63	50.6
Máximo	8.5	5.35	37
Media	6.2	5.2	16.1
Varianza	2.92	0.38	86.9

disminuye en un 86.9%, lo que indica que han disminuido considerablemente las ráfagas durante la transmisión.

Con el objetivo de evaluar cuanto afecta a la calidad de vídeo recibido la transcodificación, extrajimos la VQ de ambos vídeo utilizando el programa *Eleccard Video Quality Estimator*. El valor medio obtenido es de 4.3 sin transcodificación y 4.2 con transcodificación. Por tanto, aunque se haya reducido un poco la calidad durante este proceso, el resultado continúa siendo altamente satisfactorio en base a la recomendación (ITU-T/BT.500 2012).

4.4.1.2 *Aplicación de transcodificación en un problema de congestión durante una transmisión IPTV*

Para finalizar esta sección presentaremos un ejemplo práctico de aplicación de transcodificación en un sistema de gestión de IPTV. Para ello, planteamos el siguiente problema. Tenemos una red en la que se está llevando a cabo una distribución de servicio de IPTV. En la red se está produciendo congestión por falta de ancho de banda al entregar el servicio a varios clientes, tal como se observa en la figura 4.35. Este problema está provocando una disminución de la QoE del usuario.

Como se puede observar en la figura 4.35, tenemos 6 usuarios que pueden disfrutar de un servicio de IPTV en multicast. Si el tamaño medio del vídeo transmitido, según la tabla 4.9, es de 6,2Mbps, necesitaríamos en la red de distribución 37,2 Mbps. El enlace entre la red central MPLS y el router ADSL de distribución es un T-3 a 45Mbps. Si se conectaran todos los usuarios a la vez la QoE se degradaría rápidamente debido a un error de estimación del ancho de banda ya que deberíamos de haber tomado en cuenta posibles valores máximos que podrían darse y no únicamente el medio, ya que según la tabla 4.9 podrían darse picos de 8.5Mbps, sobrepasando así el ancho de banda disponible. Como consecuencia se denegaría

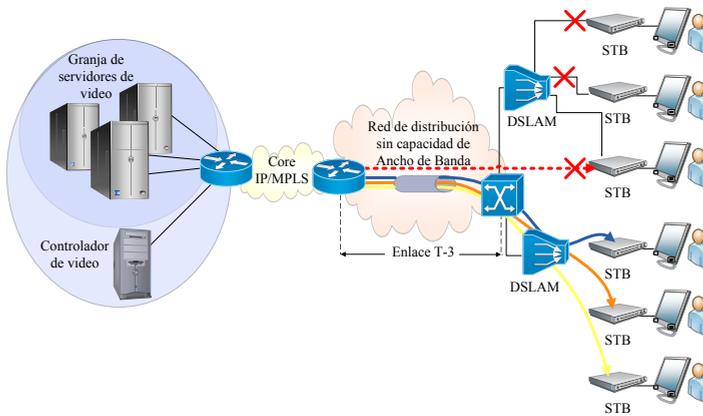


Figura 4.35: Situación de congestión en la red por falta de ancho de banda durante el acceso al servicio IPTV.

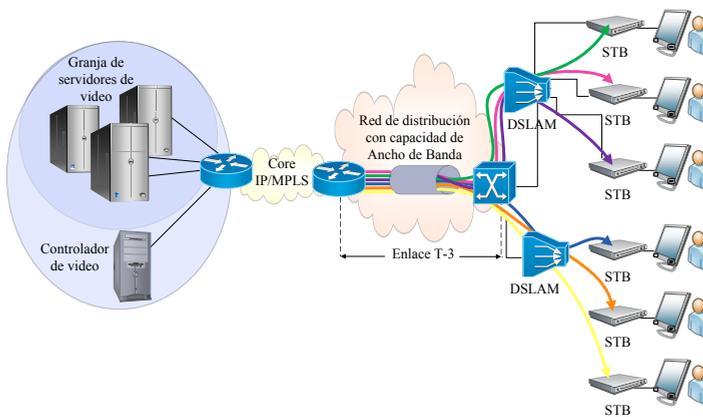


Figura 4.36: Solución de la congestión con la agrupación de grupos multicast.

el servicio tal como se puede observar en la figura 4.35. Si dispusiéramos de un sistema de gestión basado en transcodificación, en esta situación se realizaría una disminución de la tasa de bits del flujo de vídeo por ejemplo un 16,1 %, según la tabla 4.9. De esta forma se daría servicio a todos los usuarios tal como se puede observar en la figura 4.35.

Para ampliar la información de este ejemplo de aplicación se puede consultar el trabajo (Lloret y col. 2011b).

4.5 Conclusiones

En este capítulo hemos analizado algunos de los factores que influyen en la QoE durante la transmisión de un flujo de vídeo para IPTV. Estos factores son: el handover, los formatos de codificación, la cuantificación y la transcodificación. Hemos llevado a cabo medidas preliminares para ver cómo afectan.

El handover es un proceso muy importante en la transmisión de flujo IPTV ya que puede afectar a la QoE. Este proceso ha sido analizado mediante un caso de estudio. Las conclusiones tras las mediciones demuestran que es factible que se produzca handover durante una transmisión de flujo IPTV, ya que aunque durante el proceso de itinerancia se han producido más pérdidas de paquetes, un aumento del retardo y del jitter, no han supuesto el suficiente impacto en la calidad del vídeo recibido por el usuario para que no se pueda dar. Hemos observado también que los casos más graves se dan con tecnologías IEEE 802.11a. Para el desarrollo de esta sección hemos hecho referencia a una de nuestras publicaciones, en concreto a (Canovas y col. 2012), basada en el análisis del proceso de handover en transmisiones de flujo de vídeo.

Otros aspectos que hemos tratado en este capítulo han sido los formatos de compresión y contenedores. Para definirlos los hemos agrupado en: estándares de compresión MPEG y serie ITU H-XXX y formatos contenedores, formatos de compresión y contenedores para dispositivos móviles y formatos de codificación para 3D/ST. Hemos analizado la tasa de compresión y el ancho de banda utilizado para cada uno de estos grupos. En base a los resultados obtenidos hemos extraído las siguientes conclusiones. Entre el grupo formado por los formatos MPEG y serie ITU H-XXX, el formato de mayor tasa de compresión es MPEG-4 el cual tiene un factor de compresión considerablemente superior al resto. También es el formato que menos ancho de banda utiliza. En cuanto a los formatos para dispositivos móviles diremos que aquellos formatos de codificación móvil que utilizan contenedores 3GP y 3G2 han dado muy buenos resultados en todos los experimentos que hemos realizado. También hemos concluido que el formato utilizado en Android no es recomendable en transmisiones en las que se necesite consumir poco ancho de banda o se necesite una baja tasa de bits. Por último, hemos comparado el formato

2D+Z con el estereoscópico y hemos comprobado que 2D+Z tiene una mayor tasa de compresión. Los datos extraídos para llevar a cabo esta comparación son de uno de nuestros estudios publicado en (Lloret y col. 2011a).

Dos procesos que hemos analizado en este capítulo han sido: el proceso de cuantificación y el proceso de transcodificación. Tras su definición hemos realizado una serie de casos de estudio para comprobar aspectos como: cómo afecta el QS a la VQ, cómo afecta el QS a las condiciones de red y cuál es la importancia de la transcodificación en transmisiones de IPTV. Para el desarrollo de ambas secciones y los correspondientes casos de estudio nos hemos apoyado en algunas de nuestras publicaciones como son: (Lloret y col. 2011b) (Canovas y col. 2010b) para cuantificación y (Atenas y col. 2010) para transcodificación.

Como conclusión al primer caso de estudio diremos que el QS influye en el retardo durante la transmisión de un flujo de vídeo, así como en el número de paquetes necesarios para dicha transmisión. A menor QS menor retardo pero mayor número de paquetes. El mayor número de paquetes y el retardo entre ellos en una transmisión de IPTV será importante ya que los dispositivos intermedios necesitarán más recursos para dar el mismo servicio.

En base al segundo caso de estudio realizado se puede concluir que los formatos de codificación con factor de escala de QS=4 y QS=6 tanto para MPEG2 como para MPEG4 por lo general dan muy buenos resultados en todas las condiciones de red. Por tanto, son valores a tener en cuenta en procesos de transcodificación durante las transmisiones de flujo de vídeo IPTV cuando las condiciones de red a nivel de jitter, retardo, paquetes perdidos y ancho de banda no son las idóneas. También se ha podido comprobar como valores superiores a QS=10 son desaconsejables utilizar en transmisiones de flujo de vídeo IPTV ya que sobre todo la calidad de vídeo que ofrecen es muy baja.

Por último, como conclusión al tercer caso de estudio, podemos decir que con la utilización de la transcodificación en transmisiones de IPTV se ha planteado una técnica con la que poder solucionar situaciones de congestión de red y por tanto de degradación de la QoS. Resaltar que una disminución en la QoS puede afectar a la QoE. También podemos considerar esta técnica de transcodificación como una alternativa a la catalogación de tráfico (traffic shaping). La catalogación de tráfico consiste en retrasar paquetes para optimizar o garantizar el rendimiento de la red. La catalogación de tráfico provoca retardo, lo que afecta directamente a la QoS de red. Esto hace que se considere aún más la transcodificación como una buena alternativa en sistemas de IPTV. Adaptar la transcodificación a un sistema de gestión para garantizar la QoE del usuario final como un método proactivo de control de la congestión o como un método para reducir la tasa de bit en canales de IPTV es una de las bases de nuestro trabajo de investigación.

Capítulo 5

Sistema Inteligente de gestión

En este capítulo desarrollaremos los aspectos teóricos del funcionamiento del sistema de gestión. En la sección 5.1 introduciremos el tema. En la sección 5.2 llevaremos a cabo el desarrollo matemático de los métodos inductivos de predicción que utilizaremos en el sistema de gestión para predecir la QoE. Por último, en la sección 5.3 llevaremos a cabo el desarrollo matemático del sistema de gestión basado en los métodos inductivos de predicción.

5.1 Introducción

El sistema de gestión inteligente planteado va a necesitar un método inductivo de predicción. Debemos conocer la QoE de un usuario para valorar la necesidad de ajustar los parámetros de red o en su defecto el formato de codificación de la transmisión. Trataremos de estimar este valor a partir de los parámetros de QoS. Es decir, necesitamos una función que tome como entrada el jitter medio, el retardo medio, los paquetes perdidos, el ancho de banda y el códec de transmisión; y que obtenga una estimación de la QoE como salida. Para obtener esta función aplicaremos un método inductivo. Es decir, aprenderemos la función a partir de un corpus de entrenamiento y lo validaremos con uno de test.

Las redes neuronales artificiales, según (Bishop 1995) y (Haykin 1994), constituyen una de las áreas de la ingeniería del conocimiento que más se han desarrollado en los últimos años. Sus propiedades y características han hecho de ellas una herramienta usada con frecuencia en la resolución de problemas reales de gran complejidad. Se han aplicado en diferentes áreas, como por ejemplo, el reconocimiento de patrones, diagnóstico médico, modelado de datos financieros, predicción de datos de crecimiento microbiano, etc. En este trabajo de investigación nos encontramos ante un

problema de estimación de una función. Para este tipo de problemas las redes neuronales (RN) que se suelen utilizar son redes neuronales de regresión (RNR). Una de las funciones de activación de las RN de regresión son las unidades producto (RNRUP). Una estructura utilizada con frecuencia en RNRUP es la formada por una capa oculta con unidades producto y una capa de salida con un nodo tipo sigmoide que se utiliza como clasificador.

En el área de aplicación de las RN, uno de los campos más interesantes es de la regresión no lineal (Hwang y col. 1994). El hecho de que una red neuronal con funciones de transferencia sigmoideal pueda aproximar cualquier función continua dada con la precisión deseada (Hornik, Stinchcombe y White 1989) es un fundamento de peso para la aplicación de RNR. Sin embargo, debido a la falta de transparencia de las funciones sigmoideas en varios problemas de regresión, se ha propuesto diferentes funciones como alternativa. Entre otras, podemos mencionar las funciones de Gauss (Kosko 1992), las funciones de base radial (Lee y Hou 2002) y de regresión general (El-Melegy 2014). De la misma manera, recientemente se ha propuesto nuevos modelos de redes para la regresión y clasificación (Arulampalam y Bouzerdoum 2003).

Las redes RNRUP han demostrado buenos resultados en el modelado de datos en los que existen interacciones de diferente orden entre las variables independientes del problema. Las redes neuronales multiplicativas contienen nodos que multiplican sus entradas en lugar de agregarlas o sumarlas. Por tanto permiten entradas para interactuar no linealmente. Se ha comprobado la eficiencia en el modelado de determinado tipo de datos usando un número de nodos en la capa intermedia, inferior al que se necesitaría si se consideraran redes neuronales estándar. En (Durbin y Rumelhart 1989) demostraron que la capacidad de información de una única unidad de tipo producto (medida como la capacidad para el aprendizaje de patrones booleanos aleatorios) es aproximadamente igual a $3N$, comparado con el valor de $2N$ que corresponde a una unidad de tipo aditivo, siendo N el número de entradas de la unidad. Por este motivo, se necesitan menos unidades de procesamiento que si se utilizan unidades de tipo aditivo para aproximar funciones usando unidades producto. En (Hervás-Martínez y col. 2006) también justifican el uso de redes neuronales basadas en unidades producto frente a las Sigmoid con el fin de mejorar la predicción del crecimiento microbiano.

Una buena alternativa a las RNR son las máquinas de soporte vectorial para regresión (MSVR). Existe un trabajo muy interesante en el que se compara ambos métodos para predecir la velocidad del tráfico en la ciudad en Texas (Vanajakshi y Rilett 2004). Los resultados obtenidos son muy similares. En este trabajo de investigación, también aplicamos la MSVR como posible solución al problema de estimación de una función que se nos plantea.

Las Máquinas de Soporte Vectorial (MSV) son una moderna y efectiva técnica de Inteligencia Artificial (IA) (Vapnik 1995), que ha tenido un formidable desarrollo

en los últimos años. Uno de los conceptos fundamentales en esta técnica es el algoritmo de soporte vectorial (SV), desarrollado en la Rusia en los años sesenta (Alex J. Smola 2004). El desarrollo de los vectores soporte trae consigo el surgimiento de las MSV.

Las MSV se desarrollaron inicialmente para solucionar problemas de clasificación, pero se han ampliado para problemas de regresión (Drucker y col. 1997). El tipo de resultados obtenidos con el empleo de las MSV pueden ser cualitativos o cuantitativos. Para el análisis cuantitativo se emplean MSV de regresión. Dicho método es una extensión de las MSV en el que se incluyen estimadores de rangos. El empleo de estos determinan los valores que tienen ruido dentro de la predicción a través de funciones de pérdida, donde los primeros pasos en este sentido se dieron por (Bingham, Godfrey y Tukey 1967) quienes demostraron que en situaciones reales se desconoce el modelo del ruido y dista de las distribuciones supuestas. A raíz de esto, (Huber 1964) crea el concepto de estimadores robustos los cuales están determinados por funciones de pérdida. En la actualidad, las más utilizadas son: las funciones de pérdida cuadrática, las funciones de pérdida lineal y las funciones de Huber, entre otras. En este tipo de técnicas, su estructura se determina sobre la base del conjunto de entrenamiento necesitándose pocos parámetros para el mismo. Dicho entrenamiento se reduce a la solución de un problema de optimización que se reduce a un problema de programación cuadrática. Al mismo tiempo, el uso de las funciones Kernels muestran una gran eficiencia en el resultado de la predicción

5.2 Desarrollo teórico de los sistemas de predicción para el sistema de gestión

El sistema de gestión inteligente planteado se basa en un método inductivo de predicción. Tras el análisis de varios métodos decidimos evaluar dos sistemas aplicándolos a nuestro problema: RNRUP y MSVR, ya que fueron los que mejor resultado dieron.

5.2.1 Estimación basada en Redes neuronales de unidades producto

5.2.1.1 Desarrollo

El problema de regresión que pretendemos resolver puede describirse como sigue. Para este estudio la arquitectura de la red neuronal basada en unidades producto presenta las siguientes características.

- Una capa de entrada con un nodo para cada una de las variables del vector de características. Este es, el jitter medio, el retardo medio, los paquetes perdidos, el ancho de banda y el formato de codificación.
- Una sola capa oculta con varios nodos formados por unidades producto. En concreto 4.
- Una única capa de salida con un nodo de tipo lineal. El motivo de que sea lineal es para obtener una mejor interpretabilidad de los modelos tras resolver el problema de regresión. El valor de salida se corresponderá con la estimación de la QoE.
- Se añade sesgo al nodo de la capa de salida de la red neuronal propuesta.

Por tanto, la estructura de la red neuronal que utilizaremos como sistema de predicción viene definida por 5 nodos en la capa de entrada, 4 en la capa oculta con unidades producto y un nodo en la capa de salida. La red neuronal que planteamos viene representada según la figura 5.1:

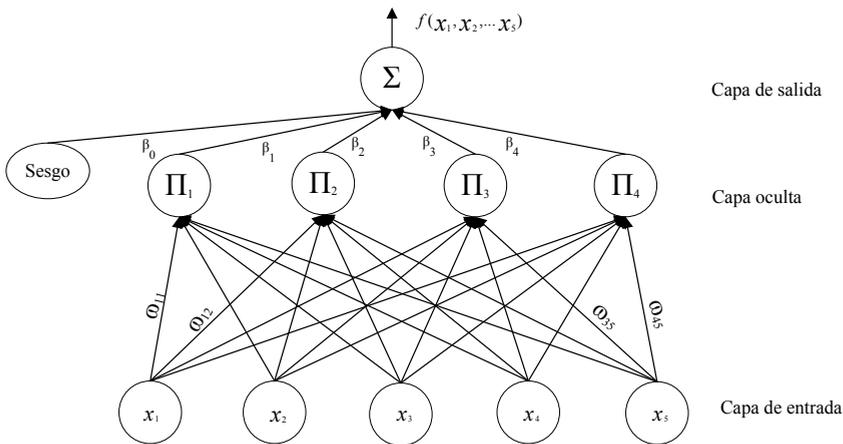


Figura 5.1: Estructura de la red neuronal basada en unidades producto

Las entradas de la red vienen representadas por las variables independientes $\{x_1, x_2, \dots, x_k\}$, donde el valor de $k = 5$. Hay $m = 4$ nodos en la capa oculta, que re-

presenta el número de términos del modelo. Existe un único nodo en la capa de salida que considerada como la única variable dependiente del modelo.

Las redes neuronales de regresión son ejemplos de modelos de regresión libres o no paramétricos. Podemos definir un problema de regresión de modelo libre de la siguiente manera (Hwang et al., 1994). Dado n pares de vectores,

$$(\vec{x}_l, y_l) = (x_{l1}, x_{l2}, \dots, x_{lk}; y_l), \quad l = 1, 2, \dots, n$$

Donde los valores y_l han sido generados a partir del siguiente modelo desconocido,

$$y_l = f(\vec{x}_l) + \varepsilon_l, \quad l = 1, 2, \dots, n$$

siendo \vec{x}_l las variables independientes del problema e y_l los valores de salida. La función f es una función desconocida definida en el espacio euclídeo de dimensión k con valores en \mathbb{R} , de forma que

$$f : \mathbb{R}^k \Rightarrow \mathbb{R}$$

ε_l , para $l = 1, 2, \dots, n$, son variables aleatorias independientes entre sí e independientes de \vec{x}_l de media cero, $E[\varepsilon_l] = 0$. El principal objetivo del problema de regresión es estimar la función f , a partir de los datos de entrenamiento (\vec{x}_l, y_l) , que mejor se ajuste a estos datos minimizando el error cuadrático medio:

$$y = f(\vec{x}) \tag{5.1}$$

La activación de cada nodo multiplicativo j de la capa oculta vienen dado por:

$$\phi_j = \prod_{i=1}^k x_i^{\omega_{ji}}, \quad \omega_{ji} \in \mathbb{R} \tag{5.2}$$

donde ω_{ji} son los pesos que conectan el nodo j de la capa oculta con el nodo i de la capa de entrada. Si los exponentes valieran $\{0, 1\}$ estaríamos obteniendo una unidad de alto orden, también conocidas con el nombre de sigma-pi. Sin embargo, estamos ante un problema en el que los exponentes no son fijos y por tanto una red de unidad producto (UP). En este caso concreto es una red combinatoria con únicamente la capa oculta con unidades producto y el resto (la salida) con unidades suma.

La activación del nodo de la capa de salida viene dada por:

$$\sum_{j=1}^m \beta_j \phi_j + \beta_0, \quad \beta_j \in \mathbb{R} \quad (5.3)$$

Siendo β_j el peso que conecta el nodo j de la capa oculta con el nodo de salida y β_0 el sesgo (bias). Obsérvese que debido a la función de transferencia de las unidades producto, aunque el modelo no llevara expresamente el sesgo, el sesgo se podría considerar como un nodo de la capa oculta que no tiene conexión con ningún nodo de la capa de entrada y con conexión con la capa de salida.

Para resolver el problema de regresión consideraremos la familia de funciones definidas a partir de redes neuronales basadas en unidades producto. Concretamente, consideraremos la familia de funciones F definidas por:

$$F = \left\{ f : A \subset \mathbb{R}^k \Rightarrow R : f(x_1, x_2, \dots, x_k) = \beta_0 + \sum_{j=1}^m \beta_j \left[\prod_{i=1}^k x_i^{\omega_{ji}} \right] \right\} \quad (5.4)$$

donde, $\beta_j \in [-M, M] \subset \mathbb{R}$, $\omega_{ij} \in [-L, L] \subset \mathbb{R}$, $i = 1, 2, \dots, k$, $j = 1, 2, \dots, m$; siendo k el número de parámetros de entrada de la función a estimar y m el número de nodos de la capa oculta (número de términos o sumandos del modelo) y siendo M y L dos constantes que nos permiten ajustar el rango de valores para las variables β_j y ω_{ij} .

El dominio de la definición de f es el conjunto de \mathbb{R}^k dado por:

$$A = \{(x_1, x_2, \dots, x_k) \in \mathbb{R}^k : 0 < x_i\} \quad (5.5)$$

Observemos que cada función de la familia puede verse como una expresión polinómica de varias variables en la que los exponentes de cada variable son números reales. En el caso más sencillo, con dos variables independientes, la función definida anteriormente vendría dada por:

$$f(x_1, x_2) = \beta_0 + \sum_{j=1}^m \beta_j x_1^{\omega_{j1}} x_2^{\omega_{j2}} \quad (5.6)$$

Como hemos definido en (5.1):

$$f(x_1, x_2, \dots, x_k) = \beta_0 + \sum_{j=1}^m \beta_j \left[\prod_{i=1}^k x_i^{\omega_{ji}} \right] \quad (5.7)$$

Esta función podría ser expresada entonces como:

$$f(x_1, x_2, \dots, x_k) = \beta_0 + \sum_{j=1}^m \beta_j \left[\prod_{i=1}^k x_i^{\omega_{ji}} \right] = \beta_0 + \sum_{j=1}^m \beta_j \exp \left(\sum_{i=1}^k \omega_{ji} \ln x_i \right) \quad (5.8)$$

Por tanto, una red neuronal basada en unidades producto puede verse desde otra perspectiva como una red neuronal con una capa oculta, donde la función de salida de cada nodo de esta capa es la función exponencial $\exp(z) = e^z$ y en donde los valores de entrada x_i han sufrido una transformación logarítmica.

5.2.1.2 *Parámetros utilizados en el algoritmo evolutivo basado en RNRUP*

En este apartado describiremos los parámetros utilizados en el proceso de aprendizaje y test para la configuración del modelo de estimación basado en redes neuronales de regresión con programación evolutiva. Para la experimentación, el conjunto de datos utilizados consta de 275 muestras, de las cuales se han tomado 250 para entrenamiento (90%) y 25 de test (10%). Cada muestra consta de 5 variables ($k = 5$) como valores de entrada y su correspondiente valor de salida. Los principales parámetros de configuración con los que hemos experimentado han sido el número de nodos de la capa oculta y las funciones de activación para cada nodo de la capa oculta. El número de nodos de la capa oculta que hemos analizado han sido valores de m entre 1 y 15. Las funciones de activación analizadas han sido: Unidad Producto y Sigmoidal. Además hemos utilizado la técnica de validación cruzada (N-fold cross validation) con valores de $N = 10$ y $N = 5$ iteraciones con el fin de estimar la precisión del modelo predictivo. En esta técnica se ha realizado por cada inicialización N subdivisiones aleatorias de los datos, donde se dispone el (90%) para entrenamiento y (10%) para test. Como criterio de parada durante proceso de aprendizaje se ha añadido un valor máximo de 1000 iteraciones.

5.2.2 **Estimación basada en maquinas de soporte vectorial para regresión**

5.2.2.1 *Desarrollo*

Un planteamiento muy similar al de RN para resolver el problema de regresión es mediante la utilización de máquinas MSVR. Siguiendo el mismo enfoque de análisis del apartado anterior, definiremos las características de la arquitectura de la MSVR como:

- Un vector de entrada compuesto por cada una de las variables del vector de características además del vector soporte. Las características son: el jitter medio, el retardo medio, los paquetes perdidos, el ancho de banda y el formato de codificación.
- Una única capa de salida con un nodo de tipo lineal. El valor de salida se corresponderá con la estimación de la QoE.

Por tanto, la estructura de la MSVR como sistema de predicción viene definida por el vector de entrada, compuesto por las características y el vector soporte, cada uno de ellos unidos por pesos a la salida tal como se puede observar en la figura 5.2:

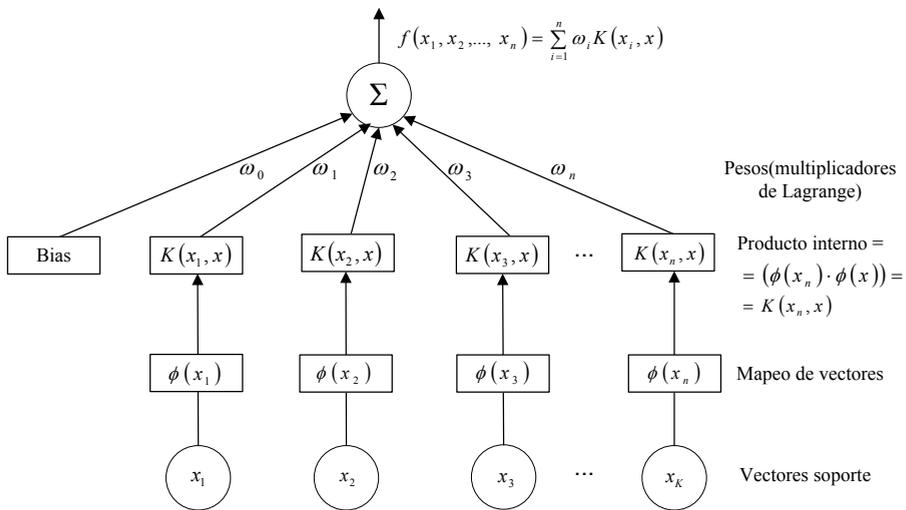


Figura 5.2: Estructura de la máquina de soporte vectorial para regresión.

Hemos utilizado una máquina MSVR con el fin de resolver el problema de regresión planteado. Para ello, el objetivo es estimar la función:

$$y = f(\vec{x}) \tag{5.9}$$

la cual ha sido definida en el apartado anterior. Esta función calcula la salida (en nuestro caso valores estimados de QoE) correspondiente al vector de entrada \vec{x} (da-

do por los vectores características compuesto por los valores de jitter, retardo, etc, de dimensión $k = 5$) de acuerdo al modelo paramétrico:

$$f : \mathbb{R}^k \Rightarrow \mathbb{R} \mid f(\vec{x}) = \langle \vec{\omega} \cdot \phi(\vec{x}) \rangle + \omega_0 \quad (5.10)$$

que como se puede observar el planteamiento es muy similar al del apartado anterior y donde,

$\langle \cdot \rangle$ es el símbolo de producto interno o producto escalar entre $\vec{\omega}$ y $\phi(\vec{x})$, $\vec{\omega}$ es el vector de pesos, ω_0 el sesgo y $\phi : \mathbb{R}^k \Rightarrow \mathbb{R}^h$ es una función que transforma el espacio de entrada de dimensión k a un espacio de características de igual o mayor dimensión h . Por tanto, el número de elementos del vector de pesos $\vec{\omega}$ pasará de tener k a h elementos.

Para realizar esta estimación utilizaremos un conjunto de entrenamiento definido por:

$$A = \{(\vec{x}_l, y_l) \mid \vec{x}_l \in \mathbb{R}^k, y_l \in \mathbb{R} \text{ para } l = 1, 2, \dots, n\} \quad (5.11)$$

Buscamos la función f que más se ajuste al conjunto de entrenamiento A . Para ello, trataremos de minimizar la función objetivo ε , que calcula el error cuadrático entre los datos de entrenamiento y la función estimada. Esta interpretación se puede observar en la figura 5.3. Como es un problema de optimización, para minimizar también el valor de $\vec{\omega}$ utilizaremos norma Euclidiana, $\|\vec{\omega}\|^2$ de tal forma que:

$$\varepsilon_A = \frac{1}{n} \sum_{l=1}^n (f(\vec{x}_l) - y_l)^2 + \frac{1}{2} \|\vec{\omega}\|^2 \quad (5.12)$$

En esta aproximación se va a permitir una pequeña desviación entre la función $f(x)$ y los datos de entrenamiento, de forma que si es menor que ϵ , el error será ignorado. Esta función de coste es conocida por $\epsilon - Vapnik$ o $\epsilon - insensible$ y su expresión viene definida por:

$$E_\epsilon(z) = \begin{cases} 0 & z < \epsilon \\ |z| - \epsilon & k \text{ en caso contrario} \end{cases}$$

El problema ahora es minimizar la función cuadrática:

$$\varepsilon_A = \frac{1}{n} \sum_{l=1}^n E_\epsilon(f(\vec{x}_l) - y_l) + \frac{1}{2} \|\vec{\omega}\|^2$$

Dada la función paramétrica $f(\vec{x}) = \langle \vec{\omega} \cdot \phi(\vec{x}) \rangle + \omega_0$, el problema se puede plantear como la búsqueda de $\vec{\omega}$ y ω_0 tal que $y_l \leq \langle \vec{\omega} \cdot \phi(\vec{x}_l) \rangle + \omega_0 + \epsilon$ y $y_l \geq \langle \vec{\omega} \cdot \phi(\vec{x}_l) \rangle + \omega_0 - \epsilon$ para $l = 1, 2, \dots, n$

En la práctica, no siempre es posible aproximar todos los datos con un error inferior a ϵ , por lo que será necesario permitir ciertos errores introduciendo las variables $\xi_l, \hat{\xi}_l \geq 0$ para $l = 1, 2, \dots, n$ sujeto a las siguientes restricciones:

$$y_l \leq \langle \vec{\omega} \cdot \phi(\vec{x}_l) \rangle + \omega_0 + \epsilon + \xi_l \quad \text{para } l = 1, 2, \dots, n \quad (5.13)$$

$$y_l \geq \langle \vec{\omega} \cdot \phi(\vec{x}_l) \rangle + \omega_0 + \epsilon - \hat{\xi}_l \quad \text{para } l = 1, 2, \dots, n \quad (5.14)$$

$$\xi_l \geq 0 \quad \text{y} \quad \hat{\xi}_l \geq 0 \quad \text{para } l = 1, 2, \dots, n \quad (5.15)$$

donde, ξ_l y $\hat{\xi}_l$, figura 5.3, son las variables de holgura que representan las restricciones superior e inferior en las salidas del sistema.

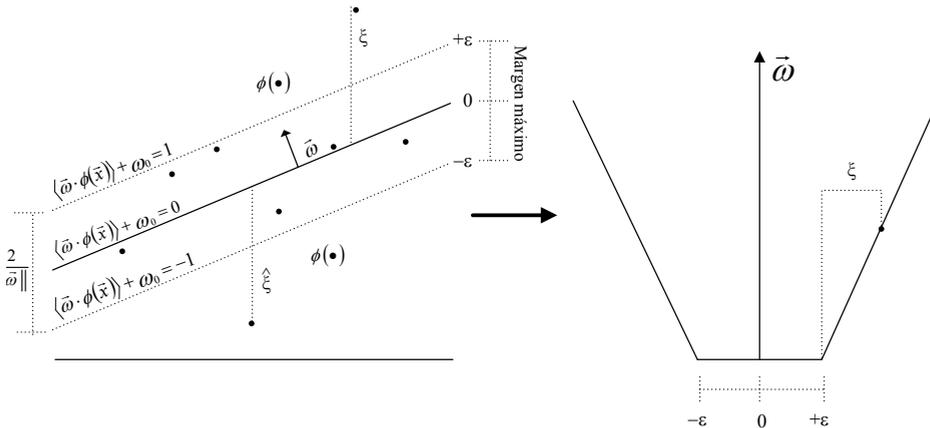


Figura 5.3: Función pérdida para la dispersión de los vectores soporte en un ejemplo de regresión lineal.

Como acabamos de ver, con la función pérdida propuesta por *Vapnik* basada en una aproximación de la función pérdida de *Huber*, mostrada en la figura 5.3, hemos obtenido una mayor dispersión en los vectores soporte. Teniendo en cuenta las variables de holgura ξ_l y $\hat{\xi}_l$, la SVR queda formulada como el siguiente problema de minimización cuadrática:

$$\varepsilon_A(\vec{\omega}, \omega_0, \vec{\xi}, \hat{\xi}) = C \sum_{l=1}^n (\xi_l + \hat{\xi}_l) + \frac{1}{2} \|\vec{\omega}\|^2 \quad (5.16)$$

donde, C es el factor de ponderación entre el error medio medido sobre el conjunto de evaluación y el error medio medido sobre un límite superior al dado sobre el conjunto de evaluación (entre riesgo empírico - riesgo estructural). (Siendo esta una de las cualidades que hace que de mejores resultados que las RN). C es por tanto una constante que determina el margen dado por ξ_l y $\hat{\xi}_l$. Cuanto mayor sea el valor de C más estricto será la $MSVR$ en permitir errores.

El funcional (3.65) se transforma, introduciendo los multiplicadores de *Lagrange* $\vec{\lambda}$ y $\vec{\gamma}$, así los datos de entrenamiento sólo aparecerán en forma de una combinación de vectores y la resolución del problema se podrá hallar resolviendo el problema dual de *Wolfe* como veremos más adelante:

$$\begin{aligned} L(\vec{\omega}, \omega_0, \vec{\xi}, \hat{\xi}, \vec{\lambda}, \vec{\gamma}, \vec{\gamma}) &= C \sum_{l=1}^n (\xi_l + \hat{\xi}_l) + \frac{1}{2} \|\vec{\omega}\|^2 \\ &- \sum_{l=1}^n \lambda_l (\epsilon + \xi_l + \langle \vec{\omega} \cdot \phi(\vec{x}_l) \rangle + \omega_0 - y_l) \\ &- \sum_{l=1}^n \hat{\lambda}_l (\epsilon + \hat{\xi}_l - \langle \vec{\omega} \cdot \phi(\vec{x}_l) \rangle - \omega_0 + y_l) \\ &- \sum_{l=1}^n \gamma_l \xi_l - \sum_{l=1}^n \hat{\gamma}_l \hat{\xi}_l \quad (5.17) \end{aligned}$$

Este resultado debe ser minimizado respecto a $\vec{\omega}, \omega_0, \vec{\xi}_i$ y maximizado respecto a $\vec{\lambda}$ y $\vec{\gamma}$. En la solución, las derivadas parciales de L respecto a las variables $\vec{\omega}, \omega_0, \vec{\xi}_i$ deben anularse:

$$\partial L / \partial \vec{\omega} = 0 \Rightarrow \vec{\omega}^* = \sum_{l=1}^n (\lambda_l - \hat{\lambda}_l) \phi(\vec{x}_l) \quad (5.18)$$

$$\partial L / \partial \omega_0 = 0 \Rightarrow \sum_{l=1}^n (\lambda_l - \hat{\lambda}_l) = 0 \quad (5.19)$$

$$\partial L / \partial \xi_l = 0 \Rightarrow \lambda_l + \gamma_l = C \quad (5.20)$$

$$\partial L / \partial \hat{\xi}_l = 0 \Rightarrow \lambda_l + \hat{\gamma}_l = C \quad (5.21)$$

Sustituyendo (3.70), (3.71), (3.72) y (3.73) en (3.69) se llega al dual de *Wolfe*, que debe ser maximizado respecto a los multiplicadores de *Lagrange* $\vec{\lambda}$:

$$L_D(\vec{\lambda}, \vec{\lambda}) = -\frac{1}{2} \sum_{l,j=1}^n (\lambda_l - \hat{\lambda}_l)(\lambda_j - \hat{\lambda}_j) \langle \phi \vec{x}_l \cdot \phi \vec{x}_j \rangle - \epsilon \sum_{l=1}^n (\lambda_l - \hat{\lambda}_l) \sum_{l=1}^n (\lambda_l - \hat{\lambda}_l) \quad (5.22)$$

Sujeto a las restricciones para $l = 1, 2, \dots, n$:

$$\begin{aligned} \lambda_l &\geq 0 & \gamma_l = C - \lambda_l &\geq 0 \\ \hat{\lambda}_l &\geq 0 & \hat{\gamma}_l = C - \hat{\lambda}_l &\geq 0 \end{aligned}$$

Además de las restricciones lineales (3.66), (3.67) y (3.68), la solución debe cumplir las condiciones de *KKT* (Karush-Kuhn-Tucker) para $l = 1, 2, \dots, n$, en la que los puntos son diferenciables en *:

$$\begin{aligned} \lambda_l^* (\epsilon + \xi_l^* + \langle \vec{\omega}^* \cdot \phi(\vec{x}_l) \rangle) + \omega_0^* - y_l &= 0 \\ \hat{\lambda}_l^* (\epsilon + \hat{\xi}_l^* - \langle \vec{\omega}^* \cdot \phi(\vec{x}_l) \rangle) - \omega_0^* + y_l &= 0 \\ (C - \lambda_l^*) \xi_l^* &= 0 \\ (C - \hat{\lambda}_l^*) \hat{\xi}_l^* &= 0 \end{aligned}$$

Con el análisis (*KKT*) se demuestra que la gran mayoría de los coeficientes de *Lagrange* son cero, y que sólo pueden ser distintos de cero para los vectores de soporte, puntos que se encuentran exactamente a la distancia marcada por el margen. Al dualizar el modelo de maximización del margen se transforma en un problema de minimización de una función cuadrática convexa sujeta a restricciones lineales. Quedando:

$$\vec{\omega}^* = \sum_{l=1}^n (\lambda_l^* - \hat{\lambda}_l^*) \phi(\vec{x}_l)$$

El conjunto de vectores soporte viene definido por:

$$SV = \left\{ \vec{x}_l \mid (\vec{x}_l, t_l) \in A : \lambda_l^* \neq 0 \text{ ó } \hat{\lambda}_l^* \neq 0 \right\}$$

Todos los puntos $\lambda_l^* = \hat{\lambda}_l^* = 0$ están dentro del tubo.

Para ω_0^* , y sujeto a la condición de dado un punto $\vec{x}_l : 0 < \lambda_l^* < C$ (análogo con $0 < \hat{\lambda}_l^* < C$):

$$\omega_0^* = y_l - c - \sum_{j=1}^n (\lambda_j^* - \hat{\lambda}_j^*) \langle \phi(\vec{x}_l) \cdot \phi(\vec{x}_j) \rangle = y_l - c - \sum_{j=1}^n (\lambda_j^* - \hat{\lambda}_j^*) K(\vec{x}_l, \vec{x}_j)$$

donde $K_{ij} = K(\vec{x}_i, \vec{x}_j)$ para $l = 1, 2, \dots, n$:

Este problema es cuadrático y convexo, por lo que la convergencia al mínimo global está garantizada usando técnicas de programación cuadrática. El problema de optimización queda expresado únicamente en función de los productos escalares $\langle \phi(\vec{x}_l) \cdot \phi(\vec{x}_j) \rangle$, los cuales se pueden evaluar mediante la función de kernel $K(\vec{x}_l, \vec{x}_j)$. Sustituyendo la expresión de $\vec{\omega}^*$ (3.70) en (3.63) se llega a la salida de la *MSVR* cuya función estimada es:

$$f(\vec{x}) = \sum_{l=1}^n (\lambda_l^* - \hat{\lambda}_l^*) \langle \phi(\vec{x}_l) \cdot \phi(\vec{x}) \rangle + \omega_0^* = \sum_{l=1}^n (\lambda_l^* - \hat{\lambda}_l^*) K(\vec{x}_l, \vec{x}) + \omega_0^* \quad (5.23)$$

El vector $\vec{\omega}^*$ admite una expansión en términos de los vectores de entrenamiento en el espacio transformado. Sólo aquellas muestras cuyos multiplicadores asociados $\hat{\lambda}_l^*$ sean distintos de 0 contribuyen a la definición de la frontera de decisión, por ese motivo reciben el nombre de vectores soporte. En base al valor de ϵ podemos distinguir los siguientes casos:

- Muestras con un error menor a ϵ : $|y_l - f(\vec{x}_l)| < \epsilon \Rightarrow \hat{\lambda}_l^* = 0$
- Muestras con un error igual a ϵ : $|y_l - f(\vec{x}_l)| = \epsilon \Rightarrow 0 < \hat{\lambda}_l^* < C$
- Muestras con un error mayor a ϵ : $|y_l - f(\vec{x}_l)| > \epsilon \Rightarrow \hat{\lambda}_l^* = C$

Debe tenerse en cuenta que solo uno de los multiplicadores de *Lagrange* puede ser distinto de 0, ya que $\lambda_l^* - \hat{\lambda}_l^* = 0$ para $l = 1, 2, \dots, n$

Por último, el parámetro ϵ permite controlar la complejidad de la solución (3.84). A medida que aumenta su valor, habrá un mayor número de muestras de entrenamiento dentro del intervalo $[-\epsilon, \epsilon]$. Los multiplicadores de *Lagrange* correspondientes a estas muestras se anulan, de modo que no intervienen en la expresión de la salida de la *MSVR*. No obstante, debe tenerse en cuenta que una solución excesivamente dispersa puede ocasionar un aumento del error.

5.2.2.2 *Parámetros utilizados en el algoritmo evolutivo basado en MSVR*

En este apartado describiremos los parámetros utilizados en el proceso de aprendizaje y test para la configuración del modelo de estimación basado en máquinas de soporte vectorial para regresión. Para la experimentación, el conjunto de datos consta de 275 muestras, de las cuales se han tomado 250 para entrenamiento (90%) y 25 de test (10%). Cada muestra consta de $k = 5$ variables independientes como valores de entrada al sistema y su correspondiente etiqueta como salida. Las funciones Kernel analizadas han sido: Polinomial y función de base radial (Radial Basis Function (RBF)). Además hemos utilizado la técnica de validación cruzada (N-fold cross validation) con valores de $N = 10$ y $N = 5$ iteraciones con el fin de estimar la precisión del modelo predictivo. En esta técnica se ha realizado por cada inicialización N subdivisiones aleatorias de los datos, donde se dispone el (90%) para entrenamiento y (10%) para test. Como criterio de parada durante proceso de aprendizaje se ha añadido un valor máximo de 1000 iteraciones.

Los parámetros de aprendizaje y optimización con los que se ha experimentado han sido:

- El valor del factor de ponderación C junto al valor del ϵ han sido los parámetros que hemos considerado de mayor relevancia en el proceso de construcción de la MSVR. C es una opción de aprendizaje. Una mala elección de C penalizará en el resultado, (Kecman 2001, página 182). En la selección del valor del parámetro C hemos tenido en cuenta una serie de consideraciones. Por ejemplo, un valor pequeño de C incrementará el número de errores de entrenamiento, mientras que un valor grande de C hace que el comportamiento se asimile a una SVM de margen estricto (Hard Margin SVM) pudiendo llegar a una situación de sobreaprendizaje (Joachims 2012, page 40). Por tanto un problema de regresión linealmente se podría solucionar con un valor de C pequeño. Para un problema más complejo como es el que nos encontramos necesitaremos un C más grande. Los valores de experimentación que hemos analizado para este parámetro han sido entre 1 y 150. A partir de este valor las pruebas de predicción empezaban a tener errores.
- El valor del factor de ϵ es el otro de los parámetros importantes de configuración. ϵ es una opción de optimización. El valor de ϵ afecta al número de vectores soporte, por tanto a la complejidad y a la capacidad de generalización de la red (Horváth 2003, página 392). Un aumento de ϵ implica una reducción de los requerimientos para la precisión de la estimación y también disminuye el número de vectores soporte. Luego mediante ϵ determinamos el nivel de precisión de la función de estimación (Kecman 2001, páginas 182-183). Tanto el valor de C como el de ϵ afecta a la complejidad del modelo. Los valores de experimentación que hemos analizado para este parámetro han sido entre 0.0001 y 1.

- Hemos definido la función kernel como $K(\vec{x}_i, \vec{x}_j)$. Dependiendo del kernel que utilizemos esta función queda definida como:
 - **Kernel Polinomial** [$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \vec{x}_j + \omega_0^*)^d$]: El principal parámetro con el que hemos experimentado en la función kernel Polinomial es con el valor d correspondiente al grado del polinomio. Hemos experimentado con valores desde 1 hasta 8 para diferentes valores de ϵ para optimización y diferentes valores de C para aprendizaje.
 - **Kernel RBF** [$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$]: El principal parámetro con el que hemos experimentado en la función kernel RBF es con el valor γ . Hemos experimentado con valores entre 0.0001 hasta 1 para diferentes valores de ϵ para optimización y diferentes valores de C para aprendizaje.

5.3 Sistema de gestión basado en métodos de predicción

El mejor resultado de experimentación se ha obtenido con redes neuronales. Partiendo de esta premisa desarrollaremos los aspectos teóricos del sistema de gestión. Utilizando la red neuronal definida estimaremos la QoE según unos valores de entrada definidos por el jitter medio, el retardo, los paquetes perdidos, el ancho de banda y el códec. En base a este proceso de estimación el sistema de gestión queda planteado de la siguiente forma.

Cuando se está produciendo una pérdida de QoE, se ha de generar una alarma. Para determinar esta situación se obtendrá una estimación de la QoE aplicando,

$$f(x_1, x_2, \dots, x_k) = \sum_{j=1}^m \beta_j \left[\prod_{i=1}^k x_i^{\omega_{ji}} \right] \quad (5.24)$$

Por otra parte, cuando el servidor recibe una alarma es posible que no pueda resolver la situación cambiando la configuración de red y vea que es imprescindible cambiar la codificación. En este caso la función que acabamos de estimar también puede ser de ayuda al sistema para determinar el códec más adecuado para las actuales condiciones de red. Por tanto, dadas las condiciones actuales de red buscamos aquel códec que maximice el valor estimado de la QoE, es decir:

$$z = \arg \max_{z \in Z} f(x_1, x_2, \dots, x_4, z) \quad (5.25)$$

siendo $Z = z_1, z_2, \dots, z_p$, el conjunto formado por todos los códecs a evaluar.

5.4 Conclusiones

En este capítulo hemos llevado a cabo el desarrollo teórico de los métodos inductivos de estimación que utilizaremos en el sistema de gestión. Se nos plantea un problema de regresión en el que dado unos parámetros de entrada al sistema de valores de la QoS estimaremos el valor de salida correspondiente a la QoE. Uno de estos métodos se basa en redes neuronales y el otro en máquinas de soporte vectorial.

En la arquitectura del método basado en redes neuronales hemos utilizado cinco nodos en la capa de entrada, cuatro nodos en la capa oculta y un nodo en la capa de salida. Hemos utilizado unidades producto como funciones de activación de los nodos de la capa oculta y función suma para la capa de salida. Por último hemos planteado cuáles han sido los parámetros de experimentación que utilizaremos en el siguiente capítulo. El parámetro que consideramos más determinante es el número de nodos de la capa oculta

El planteamiento que hemos utilizado en la máquina de soporte vectorial es el siguiente. Se mapea cada uno de los valores de entrada correspondientes a los valores de QoS y se le aplica una función Kernel. Para estimar la salida se suma cada una de estas funciones. Los parámetros utilizados en el proceso de experimentación para la extracción del modelo de estimación son: el factor de ponderación C , el factor epsilon y la función Kernel.

Por último, hemos presentado el método de estimación utilizado en el sistema de gestión para la selección del formato de codificación en una transmisión de flujo de IPTV. Este sistema se basa en buscar aquel códec que maximice el valor de QoE dadas unas condiciones de entorno de red utilizando los métodos anteriormente descritos.

Cabe decir que en este capítulo hemos hablado de resultados experimentales, sin embargo, los experimentos no han sido mostrados en este capítulo ya que se han realizado sobre 3D/ST y este tema será tratado en el siguiente capítulo. Por tanto, el desarrollo experimental lo trataremos en profundidad en el Capítulo 6, en concreto en la sección 6.4.

Capítulo 6

Sistemas de gestión basados en QoE en entornos reales de red

En este capítulo se proponen diferentes procesos de gestión para la transmisión de señales de vídeo que tratan de maximizar la QoE. En la sección 6.1 explicaremos cuál es el proceso de gestión en entornos cableados de red. En la sección 6.2 explicaremos cuál es el proceso de gestión en entornos inalámbricos de red. En la sección 6.3 explicaremos cuál es el proceso de gestión para dispositivos de telefonía móvil. En la sección 6.4 explicaremos cuál es el proceso de gestión en transmisiones de vídeo estereoscópico.

6.1 Sistema de gestión en entornos cableados de red

6.1.1 Introducción

Para el desarrollo del sistema de gestión el primer entorno de red que analizamos fue el cableado. Es evidente que este entorno presenta menos problemas a nivel de comunicación (por ejemplo, en redes inalámbricas está el problema de handover), entrega de servicios (por ejemplo, en entornos inalámbricos existe una mayor limitación del ancho de banda en la red de acceso), limitaciones en los recursos a nivel de hardware (por ejemplo, los móviles suelen tener menos recursos de procesamiento y memoria que los dispositivos que se conectan a la red cableada), etc. Sin embargo, al ser una de las primeras partes que tratamos en el desarrollo del sistema de gestión, es importante realizar un planteamiento firme y coherente que nos valiera para el resto del trabajo de investigación. Dos publicaciones en las que

presentamos las bases del sistema de gestión son (Lloret y col. 2011b) y (García y col. 2009).

Uno de los aspectos más importantes en el inicio del desarrollo del sistema de gestión fue la utilización de ciertos parámetros de red como medidas para estimar la QoE de forma objetiva. Para ello tuvimos que demostrar que algunos de los parámetros de QoS influyen directamente en la QoE. En (Siller y Woods 2003), los autores mejoran la QoE, garantizando los requisitos de los usuarios de forma dinámica, mediante la gestión de los parámetros de QoS. Centran su trabajo en el estudio de la influencia de los parámetros de QoS tanto a nivel de red como de aplicación con el fin de conseguir dicha mejora. Los resultados obtenidos demuestran que se puede mejorar la QoE gestionando adecuadamente las métricas de QoS. Para ello es necesario un feedback o monitorización sobre la red.

(Wang 2006) es un trabajo de mucho interés para nosotros por su propuesta. En él, los autores plantean que el sistema de gestión para los servicios de IPTV debería de estar localizado entre la arquitectura del ISP y la arquitectura del usuario. Este enfoque fue la piedra angular de nuestro sistema de gestión como se puede apreciar en la sección 3.8 en la que se explica el proceso para estimar la QoE objetivamente. En base a este planteamiento preliminar llevaremos a cabo en el siguiente apartado la explicación del algoritmo de gestión para entornos de red cableados.

6.1.2 Algoritmo de gestión

El algoritmo de gestión planteado para este entorno de red se basa en el proceso de *estimación de la QoE*. Para ello, el servidor solicita parámetros de QoS (jitter, retardo, etc) de diferentes partes de la arquitectura de red. Como se puede observar en el diagrama de flujo de la figura 6.1, a partir de los datos monitorizados de la red, el sistema estima una aproximación objetiva de la QoE. Si el resultado obtenido está dentro de un límite que garantiza la QoE del usuario, el controlador de vídeo no realiza ninguna acción, simplemente continúa con la monitorización de los datos. Si por el contrario, el resultado sobrepasa un límite con el que no se garantiza la QoE del usuario, entonces se realizarán cambios en la red para mejorar el servicio. Estas acciones pueden ser por ejemplo: aumentar el ancho de banda del canal IPTV, hacer cambios en la red de acceso, etc. Si con esto no es suficiente se realiza transcoding. Todas estas operaciones se llevan a cabo desde el controlador de vídeo situado en la cabecera de red según la arquitectura de red propuesta en la subsección 3.5.1.

Como acabamos de ver, la QoE puede ser analizada bajo dos puntos de vista: por un lado tenemos la degradación de la QoE debida a los parámetros de red y por otro tenemos la degradación de la QoE debida a parámetros del usuario. Necesitamos demostrar que existe una correlación clara entre la QoS y la QoE para justificar el uso de un método inductivo para la estimación de la QoE cuyos parámetros de

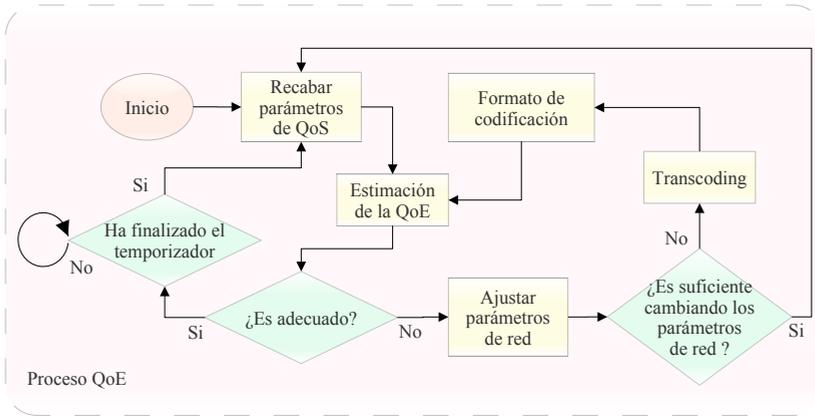


Figura 6.1: Algoritmo de gestión basado en la QoE

entrada son parámetros de QoS. En la subsección 6.1.3 se ha realizado un banco de pruebas para demostrar la influencia de los parámetros de QoS en la degradación de la QoE. Una forma de abordar una estimación de la QoE estrechamente relacionada con nuestro planteamiento es la de la ecuación (6.1), la cual ha sido presentada en (Lloret y col. 2011b).

$$QoE_{Total} = QoE_R + QoE_U = \frac{1}{t_{retardo} + K_1 t_{jitter} e^{pp}} + k_3 \frac{\log_{10}(VQ)}{t_{zap} + k_2 t_{sinc}} \quad (6.1)$$

donde,

k_1, k_2, k_3 son constantes positivas de ajuste. Con k_1 aumentamos el peso del tiempo de jitter (t_{jitter}) con respecto al tiempo de retardo ($t_{retardo}$). Con k_2 aumentamos el peso del tiempo de sincronismo (t_{sinc}) con respecto al tiempo de zapping (t_{zap}). Con k_3 aumentamos el peso de la QoE_U con respecto al de QoE_R . VQ es la calidad de vídeo recibida por el usuario, (t_{sinc}) el tiempo de sincronización entre el audio y el vídeo, (t_{zap}) el tiempo de zapping, (pp) los paquetes perdidos, ($t_{retardo}$) el tiempo de retardo y (t_{jitter}) el tiempo de jitter durante la transmisión. La VQ se calcula comparando el vídeo transmitido con el vídeo recibido. Se propone la función \log_{10} para dar más peso a los valores altos de la VQ . La escala propuesta de VQ va del intervalo de 0 a 10, siendo 10 la máxima calidad de vídeo posible recibida.

En el planteamiento anterior se intenta aproximar el valor de QoE en base a los parámetros de red (QoE_R) y a los parámetros del usuario (QoE_U). De tal forma que a medida que aumenta el valor de los parámetros de red la (QoE_T) disminuye. Podemos observar que este planteamiento es más bien conceptual sin un exhaustivo rigor matemático. La finalidad es ver qué peso tiene la (QoE_R) con respecto a la (QoE_U) en la disminución de la (QoE_T). Para ello analizaremos en el estudio realizado en subsección 6.1.3 cómo afectan los parámetros de (QoE_R) en la VQ.

Con el fin de analizar en profundidad cuánto afectan el tiempo de zapping y el tiempo de sincronismo en la (QoE_U) realizamos el estudio presentado en (García y col. 2009). De acuerdo con (S. Winkler 2007), unos los parámetros que más influyen en la (QoE_U) son:

- La calidad de audio y vídeo
- El tiempo de sincronización entre el audio y vídeo
- El tiempo de zapping

En base a estos parámetros y al estudio realizado en (García y col. 2009), podemos decir que si la calidad de vídeo es alta, es porque la señal de IPTV recibida por el usuario es buena y por tanto influirá muy positivamente en que exista una buena QoE. Tal como se puede observar en la figura 3.2, en una transmisión IPTV el tiempo medio de zapping oscila entre 2,5s y un 3,5s (depende del tamaño del GOP) y el tiempo medio de sincronización entre audio y vídeo según el (*Ministerio de Industria, Turismo y comercio*) es de $\pm 160ms$. El tiempo de zapping y de sincronismo son inversamente proporcionales a la VQ, ya que cuanto mayor sean más negativamente influyen en la QoE. En base a este análisis determinamos que la aproximación más correcta de la QoE_U , según la ecuación (6.1), es sin el valor de ajuste k_3 dando así más importancia a los parámetros de QoS.

Como hemos comentado al inicio de este apartado, si el sistema de gestión considera que la QoE ofrecida no supera un umbral, primero intentará solucionar el problema realizando cambios en la red y si no lo soluciona de esta forma realizará transcodificación sobre el vídeo que se está transmitiendo. El proceso de transcodificación está explicado con detalle en la sección 4.4. Planteamos dos formas para seleccionar el formato de codificación con el que se va a llevar a cabo la transcodificación: heurística e inteligente. La forma inteligente se basa en métodos inductivos de estimación y la trataremos con detalle en la sección 6.4. En esta sección trataremos la heurística, aunque hay que concretar que en redes cableadas el sistema de gestión puede llevar a cabo la transcodificación tanto de forma heurística como inteligente. La información ha sido extraída del estudio realizado en (Lloret y col. 2011c). El proceso para llevar a cabo la selección del formato de codificación de forma heurística se muestra en la figura 6.2.

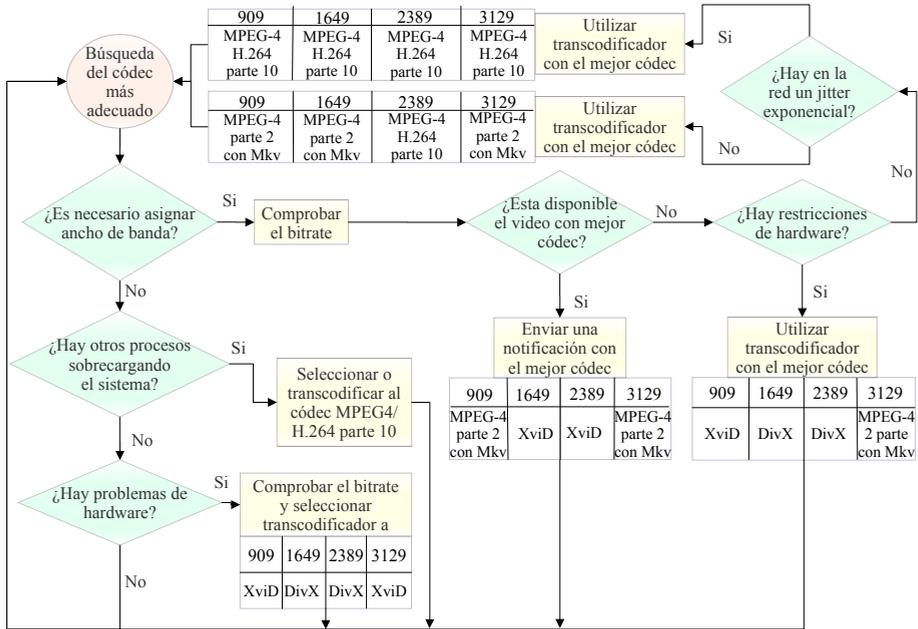


Figura 6.2: Selección del códec en base a las características del sistema.

Como vemos en la figura 6.2 el primer factor que se toma en cuenta es el ancho de banda. Por un lado, si hay restricciones de ancho de banda se comprueba la tasa de bit o bitrate del vídeo. Si hay algún códec con el que se pueda mejorar el valor actual se selecciona. Si no lo hay, se comprueba si existen restricciones de hardware por parte del usuario. En el caso de que no las haya se comprueba el jitter de la red y en base al resultado se lleva a cabo el proceso de transcodificación con el códec más adecuado. Si por el contrario existen restricciones de hardware, directamente se lleva a cabo la transcodificación con el códec más adecuado. Por otro lado, si no existen restricciones de ancho de banda se comprueba si hay procesos que estén sobrecargando el sistema. Si se está sobrecargando se selecciona el códec con el que se pueda mejorar. Si no es así, se comprueba si existen restricciones de hardware. Si existen, se selecciona el códec correspondiente y se lleva a cabo la transcodificación. Si no existen, finaliza el algoritmo sin haber encontrado un códec que se adapte a las condiciones actuales según nuestras investigaciones. Todo este proceso se puede simplificar según la figura 6.3.

Como vemos, el transcodificador básicamente comprueba si existen limitaciones en el ancho de banda, el tiempo de transcodificación o las características del dis-

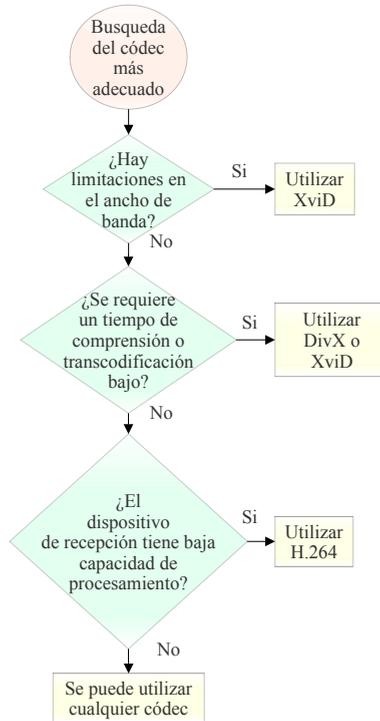


Figura 6.3: Selección del códec en base a las características del sistema simplificado.

positivo del usuario. En base a esta información determina qué códec es el más idóneo para el entorno de red en el que nos encontramos.

6.1.3 Análisis de los parámetros de QoS que influyen en la QoE

La principal finalidad de este estudio es analizar cómo los parámetros de red influyen en la calidad de vídeo recibido por el usuario y por ende en la QoE. Para llevar a cabo el test simulamos una red IPTV utilizando un servidor de vídeo VLC, un cliente VLC y el programa analizador de paquetes Wireshark en la parte del cliente tal como se ha descrito en la sección 3.9. La arquitectura de red utilizada en el banco de pruebas se ha extendido con 2 switch y 2 routers, de tal forma que el servidor IPTV está conectado a uno de los switches, este a su vez a un PC con Netdisturb, el PC a un router el cual está conectado a otro con un enlace serial y por último este router conectado a un switch al que están conectados los clien-

tes. Las características del flujo de vídeo transmitido son: codificación MPEG-2, resolución de 720x576, a 25 fps, una duración de 16 segundos y formato DVD pal.

6.1.3.1 Prueba del retardo

Para llevar a cabo esta prueba, se introdujeron diferentes valores de retardo en la red con el fin de conocer cómo afecta este parámetro en la calidad de vídeo. Los valores introducidos fueron de 5, 10, 200 y 400 milisegundos (ms). En base a los resultados obtenidos se puede observar que la variación de retardo en la red entre los 5 y 400ms en un entorno cableado no influye prácticamente en la calidad de vídeo recibida por el usuario. No hay cambios significativos en la calidad tal como se pudo observar en la figura 6.4 y la figura 6.5. En la figura 6.4 se muestra la calidad de vídeo recibida durante una transmisión con 20ms de retardo en la red y en la figura 6.5 200ms. Debemos tener en cuenta que valores superiores a 400ms afectarán directamente al tiempo de zapping lo que provocará una degradación en la QoE percibida por el usuario.

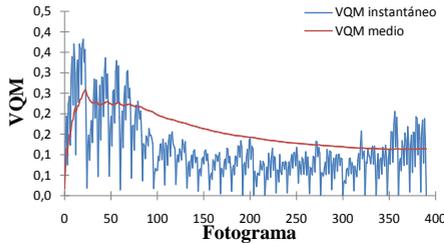


Figura 6.4: Promedio de la VQM con un retardo en la red de 20ms.

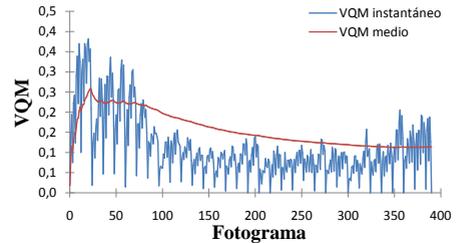


Figura 6.5: Promedio de la VQM con un retardo en la red de 200ms.

6.1.3.2 Prueba del jitter

Para el análisis del jitter configuramos una red con un retardo constante, con los mismos valores que el caso anterior, pero añadiendo dos modelos diferentes de jitter, exponencial y uniforme. Al igual que en caso anterior, en un entorno cableado aunque exista jitter en la red los cambios en la calidad de vídeo recibida por el usuario no son significativos. Además de las técnicas de buffering utilizadas actualmente en los reproductores multimedia y en los set-top-box permiten evitar en mayor medida este problema. Sin embargo, tal como veremos en la sección 6.2 y la sección 6.3, cuando el entorno es inalámbrico el jitter sí que es un factor a tener en cuenta.

6.1.3.3 Prueba de los paquetes perdidos

Para llevar a cabo esta prueba se introdujeron pérdidas de paquetes en la red durante la transmisión. Los porcentajes de pérdidas analizados fueron de: 0.01 %, 0.5 %, 2 %, 8 % and 15 %. En la figura 6.6 se muestran los resultados obtenidos. Vemos que cuanto mayor es el porcentaje de paquetes perdidos la métrica VQM aumenta, lo que significa que la calidad del vídeo empeora. La amplitud de los picos representa la cantidad de imágenes con errores durante la transmisión del vídeo, ya que cuando hay pérdida de paquetes, aparecen bloques distorsionados en las imágenes recibidas por el usuario.

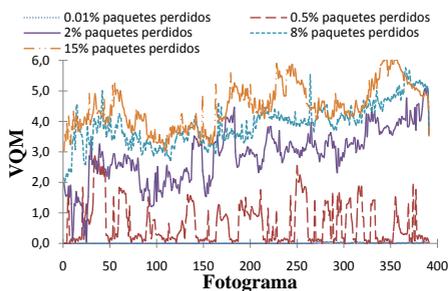


Figura 6.6: Promedio de la VQM con diferentes valores de pérdidas de paquetes en la red.

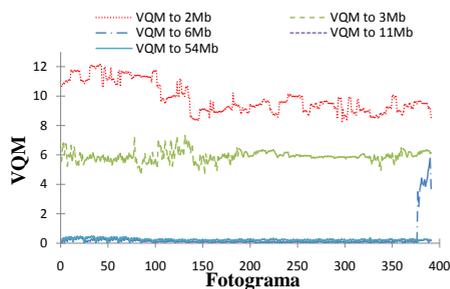


Figura 6.7: Promedio de la VQM con diferentes anchos de banda de red.

6.1.3.4 Prueba del Ancho de banda

Para llevar a cabo el estudio de la influencia del ancho de banda en la calidad de vídeo durante la transmisión de un flujo de vídeo hemos supuesto los siguientes entornos cableados: Un canal de banda ancha RDSI europeo a 2Mbps, un canal típico MPEG-2 SDTV a 3Mbps y un ancho de banda típico ofrecido por un proveedor de servicios de 6Mbps. El vídeo original requiere 6.3Mbps. Los resultados obtenidos de VQM durante la transmisión bajo dichas condiciones se muestran en la figura 6.7

En la figura 6.7 podemos observar que con 2 y 3Mbps no hay suficiente ancho de banda para transmitir un canal SDTV, ya que se producen muchas pérdidas de paquetes y la calidad de vídeo disminuye considerablemente. Con 6Mbps, la calidad de vídeo recibida durante la transmisión es variable. En las 200 primeras muestras la calidad es estable, ya que durante este periodo de tiempo la cola virtual del router está bien gestionada. A partir de ese momento la calidad empieza a empeorar ya que ha superado la capacidad máxima de memoria de la cola virtual del router, lo que provoca que se pierdan todos los paquetes nuevos entrantes, dando como resultado una mala calidad de imagen para el usuario final. Cuando

el ancho de banda es de 11Mbps no existe ningún problema durante la transmisión y la calidad de vídeo recibida es perfecta.

6.1.3.5 Prueba del tiempo de zapping

Tal como se ha analizado en la subsección 3.4.1 y como se corrobora en (Song y Kwon 2009) el tiempo de zapping es otro parámetro que podría afectar a la QoE del usuario. En (Kooij, Kamal y Brunnström 2006) se realiza un análisis de cómo estos tiempos afectan a la QoE; los autores afirman que un cambio de canal de 0.1s a 0.5s hace percibir al usuario que la respuesta del sistema es prácticamente instantánea, lo que aumenta el MOS, es decir, el servicio comienza a ser bien recibido; mientras que un tiempo de más de 1s hace que el usuario pierda la sensación de que el servicio opera continuamente lo que provoca una degradación en la QoE teniendo como consecuencia deserciones y usuarios insatisfechos.

Al cambiar de canal, si el canal no está disponible en el STB el decodificador envía un mensaje Internet Group Management Protocol (IGMP) con información del canal actual y el solicitado a la puerta de enlace. Si el router no lo tiene disponible replica el mensaje a la red y si lo tiene devuelve el flujo del canal solicitado. Este último paso se repite hasta llegar a la red troncal en el caso de que ningún router de los siguientes saltos disponga del canal solicitado. En la red troncal se disponen de todos los canales y desde aquí ya sí que se transmite seguro el flujo solicitado al STB. En todo este proceso, el cambio de canal podría influir sobre la puerta de enlace del usuario pudiendo provocar retardo y pérdida de paquetes lo que causaría también una degradación en la QoE del usuario.

Según el estudio que presentamos en la subsección 3.4.1 se observa que en redes xDSL el tiempo de zapping medio sobrepasa el segundo, por lo que es un problema que debemos considerar. El zapping está directamente relacionado con el jitter y el retardo, por tanto son dos parámetros que aunque no influyan directamente en la VQ pueden influir en la QoE por lo que los hemos tenido en cuenta en nuestro método de estimación de la QoE.

6.1.4 Conclusiones

En esta sección se ha desarrollado un sistema de gestión para entornos cableados basado en parámetros de QoS, con el fin de garantizar la QoE. Para su desarrollo, partimos de la premisa de que los parámetros de QoS influyen en la QoE. Monitorizamos los parámetros de red con el fin de diseñar un sistema de gestión basado en métodos de estimación de la QoE. Desarrollamos un algoritmo de gestión basado en la información recabada de la red. En el caso de que no se garantice la QoE, una de las soluciones propuestas por el sistema de gestión es realizar cambios en el tráfico del servicio de IPTV para evitar principalmente congestión en la red y pér-

didadas de paquetes, ya que como hemos demostrado son parámetros que influyen de forma muy directa en la QoE. En el caso de que no se pueda solucionar mediante la gestión del tráfico, una solución alternativa es realizar transcodificación.

Las principales conclusiones que podemos extraer del estudio realizado son: por un lado que el jitter en tecnologías FastEthernet es bastante estable y por otro que unos de los parámetros de QoS que más influyen en la degradación de la QoE son el ancho de banda y la pérdida de paquetes. Podemos considerar ambos parámetros estrechamente relacionados, ya que un ancho de banda bajo tiende a provocar pérdidas de paquetes. Por último, el jitter y el tiempo de zapping deben de ser considerados, especialmente en redes inalámbricas.

6.2 Sistema de gestión en entornos inalámbricos de red

6.2.1 Introducción

La evolución de las tecnologías inalámbricas se centra principalmente en el aumento de la velocidad de transferencia de los datos. Su objetivo es permitir a los nuevos servicios multimedia que requieren mayor ancho de banda funcionar correctamente y así garantizar las necesidades de los usuarios finales tal como analizamos en (Lloret y col. 2013).

El procedimiento de handover permite el acceso inalámbrico en zonas que están cubiertas por los proveedores de acceso sin necesidad de una reconexión continua. Esta necesidad de mantenerse continuamente conectado ha dado lugar a la estandarización en 2008 de la entrega independiente de medios (Media Independent Handover (MIH)) (IEEE/STD/802.21 2008). Esta estandarización favorece que los esfuerzos realizados en la gestión de las tecnologías implicadas en el handover vertical (Pontes y col. 2008) vayan en la misma dirección.

En el proceso de handover, se puede incluir distinta información para llevar a cabo decisiones en función del campo de aplicación. Por ejemplo, en (Balasubramaniam e Indulska 2004), los autores han utilizado la información de contexto con respecto a los dispositivos de usuario, la ubicación del usuario, del entorno de red y solicitudes de QoS. Otros trabajos tienen en cuenta el consumo de energía para tomar decisiones de handover como en (Joe, Kim y Hong 2007).

Es importante garantizar la recepción de datos continua y la QoS en sistemas inalámbricos móviles. En (Fodor, Eriksson y Tuoriniemi 2003) plantean la forma de garantizar la QoS mientras el conjunto de criterios y mecanismos de selección de acceso permiten al usuario seguir conectado a varios servicios de forma casi óptima. Toda esta información es muy válida para el desarrollo de nuestro sistema de gestión.

La recepción de flujo de vídeo en entornos inalámbricos es más susceptible a una degradación de la QoE que en los entornos cableados. Motivos como el número de clientes conectados, congestión en la red, falta de ancho de banda o proceso de handover, entre otros, pueden dar lugar a una QoE entregada inaceptable.

6.2.2 Algoritmo de gestión del servidor IPTV

El algoritmo de gestión propuesto se ha desarrollado para entornos de red inalámbricos con el fin de garantizar la mejor calidad posible de servicio IPTV entregada al usuario final. El diagrama de flujo se puede observar en la figura 6.8. Consiste básicamente en la siguiente serie de procesos. En primer lugar, tras conectarse el cliente IPTV de forma correcta a la red de servicios IPTV inalámbricos, el servidor IPTV monitoriza cada intervalo de tiempo la red para obtener los parámetros de QoS. A partir de estos parámetros el controlador de vídeo estima la QoE mediante el método inductivo de estimación. Si sobrepasa un cierto umbral, al ser una red inalámbrica, el sistema buscará otra red con mejores condiciones de QoS que garanticen la QoE del usuario final. Si existen redes a las que el dispositivo pueda conectarse, se estimará el valor de la QoE de cada una de esas redes y la que mejor valor de estimación obtenga será la red a la que el dispositivo deberá realizar itinerancia. Si no se encuentran redes tras un intervalo de tiempo, el sistema intenta solucionar el problema llevando a cabo cambios en la red y en su defecto realizando transcodificación.

Podemos observar que el algoritmo se divide en tres fases y son: la fase de análisis de calidad de servicio, la fase de búsqueda de red inalámbrica y la fase de decisión. A continuación describiremos dichas fases con más profundidad.

1. **Fase de análisis de calidad de servicio:** En la fase de análisis, después de que el dispositivo del cliente se conecte a la red inalámbrica, testea la red con el fin de analizar los parámetros de QoS. Estos parámetros son: el jitter, delay, paquetes perdidos y ancho de banda. A continuación, estos valores se almacenan en memoria. A partir de estos valores y con el formato de codificación con el que se está realizando la transmisión de flujo IPTV, se estima la QoE. Si el valor obtenido supera un umbral, el sistema pasará a la fase de búsqueda de una nueva red inalámbrica. En caso contrario, cada cierto intervalo de tiempo se vuelven a analizar dichos parámetros de red y se repite el proceso anterior.
2. **Fase de búsqueda de la Red inalámbrica:** En esta fase, el sistema inicia la búsqueda de una nueva red que mejore la QoE de la actual. Esta fase de búsqueda tiene un límite de tiempo a fin de evitar que el sistema permanezca en búsqueda mucho tiempo, ya que no es conveniente sobrecargar los recursos del sistema, tales como la unidad de procesamiento y la memoria. Para que esto no ocurra hemos establecido un temporizador al inicio de esta fase.

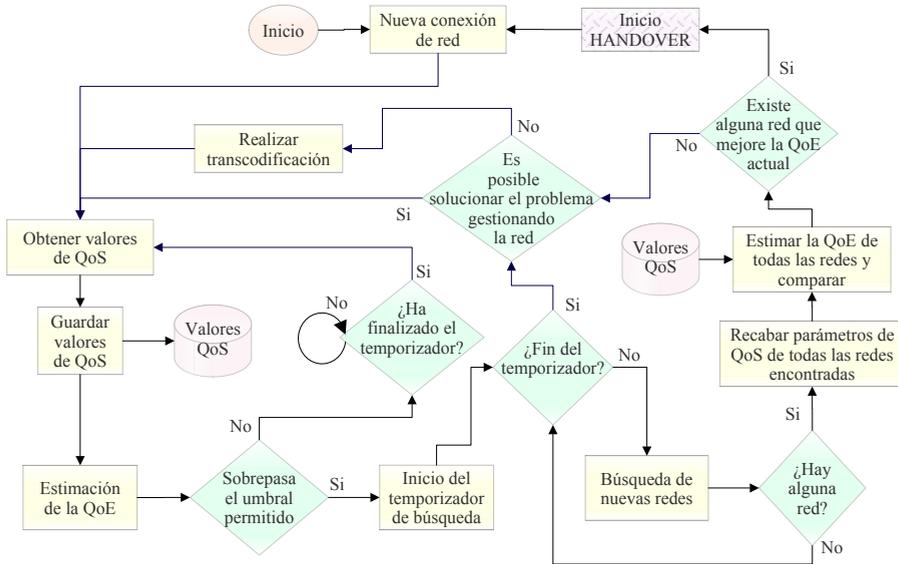


Figura 6.8: Protocolo de gestión para un servicio de distribución de red de IPTV para redes inalámbricas

Una vez que el sistema encuentra todas las redes accesibles inicia la fase de decisión. Puede ocurrir que el sistema no detecta ninguna red inalámbrica accesible y que se supere el límite de tiempo de búsqueda. Entonces pasará a la fase de decisión.

3. **Fase de decisión:** En la fase de decisión, si se ha encontrado alguna red accesible, el controlador de vídeo comparará la estimación de la QoE de las redes accesibles con la actual. Si alguna de estas redes mejora la QoE actual, se fuerza al dispositivo a iniciar el proceso de handover con el fin de unirse a la nueva red inalámbrica. El proceso de handover se ha explicado con más detalle en subsección 4.1.1. Si no hubiera ninguna red inalámbrica accesible que mejorase la QoE o simplemente no hubiera ninguna red accesible, el sistema intentaría gestionar la red con el fin de mejorar las condiciones de QoE. Si no es posible llevará a cabo entonces el proceso de transcodificación.

Como podemos ver la parte más crítica del algoritmo se produce durante el proceso de handover, pudiendo afectar claramente a la QoS durante la entrega de IPTV. Pero una vez realizado, nos aseguramos que el cliente esté conectado a la mejor

red disponible para el servicio IPTV. Una cuestión fundamental es el análisis de este proceso para comprobar cómo afecta a la QoS entregada. Este análisis se ha llevado a cabo en el caso de estudio de la subsección 4.1.2. Una vez visto cómo funciona el sistema de gestión en el servidor IPTV, analizaremos el proceso de interacción por parte del cliente con el servidor. Analizaremos cuándo se debe de llevar a cabo el proceso de handover forzado en base a la RSSI de las redes más próximas.

6.2.3 Algoritmo del cliente IPTV

Cuando un usuario solicita un servicio de IPTV, mide la RSSI de las redes Wi-reless en su área de cobertura y se une a la de mayor valor. A continuación, el cliente envía una solicitud de conexión al servidor con el fin de comprobar si hay suficiente ancho de banda disponible para poder ver el flujo de vídeo IPTV con el formato de codificación seleccionado por el servidor. Si no hay suficiente ancho de banda disponible, el dispositivo se une a la siguiente red cuyo valor RSSI sea mayor, descartando la anterior red de la lista de redes y se repite el proceso. Si finalmente no hay suficiente ancho de banda y no se encuentra ninguna red más disponible entonces se le denegaría el servicio. Si se encuentra una red con suficiente ancho de banda, el cliente solicita el flujo de vídeo al servidor de IPTV. Mientras el dispositivo está recibiendo el flujo de vídeo el servidor controlador de vídeo continuamente monitoriza los parámetros de red y estima la QoE. En el caso de que la QoE estimada no supere un cierto umbral, se compara con la QoE estimada de otros puntos de acceso. Si la QoE actual es superior a la de otros SSIDs bajo el área de cobertura, permanece en la misma red inalámbrica. Si es inferior, el servidor envía la SSID y la dirección MAC de la red inalámbrica con mayor QoE al dispositivo del cliente. Entonces, el dispositivo sale de su red inalámbrica y se une a la nueva. Todo este proceso que acabamos de describir lo podemos observar en la figura 6.9.

Si se desea ampliar la información con respecto al sistema de gestión en entornos inalámbricos de red, se puede consultar (Lloret y col. 2013).

6.2.4 Conclusiones

En esta sección hemos descrito el algoritmo de gestión, tanto en la parte del cliente como en la del servidor, para entornos inalámbricos de red. En estos entornos de red es prioritario que el cliente esté conectado a la red inalámbrica que mejor características tenga para intentar garantizar la QoE del usuario.

Como conclusión diremos que el sistema de gestión desarrollado garantiza al cliente la conexión con aquella red inalámbrica que proporcione mejor QoE en base a la QoS. Las redes inalámbricas con un mayor número de usuarios conectados

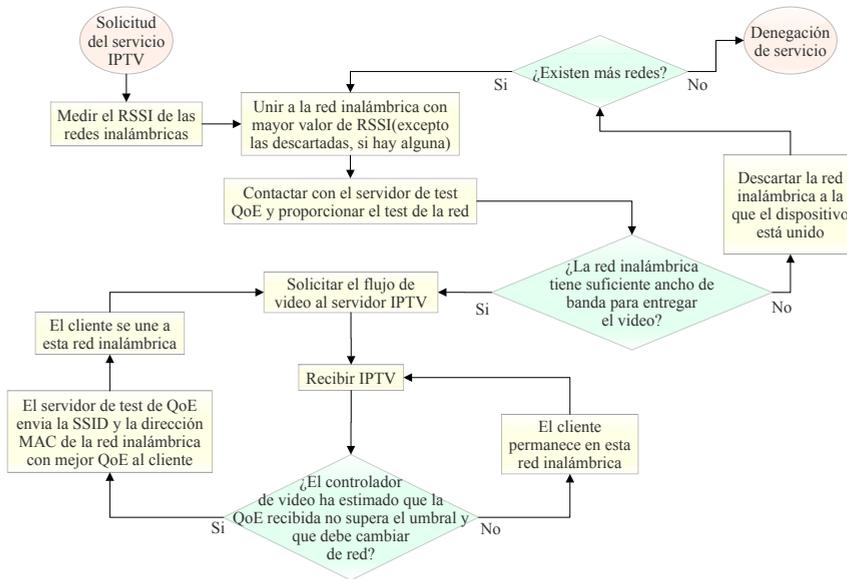


Figura 6.9: Proceso de gestión del cliente basado en QoE y RSSI para servicio de IPTV en redes inalámbricas

experimentan mayores valores de retardo, jitter y pérdidas de paquetes, lo que provoca una degradación de la QoE. En estas situaciones, el sistema de gestión equilibrará el número de usuarios conectados a cada red.

6.3 Sistema de gestión para dispositivos de telefonía móvil

6.3.1 Introducción

La TV mediante el protocolo IP apareció como un servicio multimedia que implicaba la entrega de televisión, audio, texto y datos gráficos sobre redes IP. Este servicio se extendió a redes móviles basadas en IP. Actualmente, la integración de este tipo de servicio en el mercado de la telefonía móvil ya es un hecho, debido especialmente a la mejora en la tecnología de los dispositivos móviles y a la mejora en las redes inalámbricas y de telefonía. Las redes de telefonía están evolucionando para soportar altas velocidades de transferencia de datos (Dekleva y col. 2007).

El servicio IPTV móvil da la posibilidad a los usuarios finales de una televisión en cualquier lugar y en cualquier momento. Este es un concepto atractivo para los usuarios lo que ha provocado un aumento considerable en la utilización de esta tecnología. Este hecho es importante, ya que el servicio de IPTV Móvil tiene los mismos problemas de comunicación que los teléfonos móviles, por lo que al traspaso y la entrega se refiere, tal como se apunta en (Kondrad, Bouazizi y Gabbouj 2008) y (Zeng y Agrawal 2002). Se debe considerar ambos aspectos para minimizar cualquier interrupción en la entrega de contenido, lo que provocaría que disminuyese la QoE percibida por el usuario final. Existen dos métodos para proporcionar servicio de televisión en el móvil:

- El método que utiliza dos canales de la red móvil.
- El método que utiliza un canal dedicado en modo multicast o broadcast.

Como acabamos de plantear una cuestión fundamental a tener en cuenta en un servicio IPTV móvil es la movilidad del usuario. La alta calidad del servicio IPTV recibida por parte del usuario debe lograrse incluso cuando este está en movimiento. Uno de los mayores problemas es mantener la recepción de un flujo de vídeo cuando el usuario se mueve entre las diferentes estaciones base. Por ello, se requiere un proceso de gestión que pueda asegurar la QoS y la QoE tal como plantean en (Kim y col. 2008b) para este tipo de servicio. En este trabajo se propone un sistema de gestión para IPTV móvil basado en un algoritmo de gestión de red que utiliza el ancho de banda y los parámetros de QoS con el fin de proporcionar la mejor QoE para el usuario final. Para ello, el algoritmo debe seleccionar las mejores características y la codificación adecuada del vídeo a transmitir.

6.3.2 Algoritmo de gestión de dispositivos móviles

El algoritmo de gestión de red para dispositivos móviles consta de dos partes: la parte de gestión del cliente y la parte de gestión del servidor. El algoritmo de red propuesto está pensado para tecnologías 3G y WIFI a nivel de acceso de red para una transmisión de IPTV. Sin embargo, otras tecnologías como LTE o 4G podrían ser fácilmente incluidas.

6.3.2.1 Algoritmo del servidor

El algoritmo del servidor está situado en el servidor controlador de vídeo. Dicho servidor está localizado en la cabecera de red de la arquitectura IPTV propuesta en la sección 3.5. El funcionamiento del algoritmo es el siguiente, cuando el servidor recibe una petición de canal de IPTV, si hay suficientes recursos disponibles se acepta. A continuación, el controlador de vídeo monitoriza la de red móvil con el fin de obtener los parámetros de QoS del propio enlace. Para obtener la codificación

que maximice la QoE utilizando el método inductivo de estimación, necesitamos saber, a parte del ancho de banda disponible, características del dispositivo móvil como son: la resolución de pantalla y la capacidad de hardware del dispositivo móvil, y de esta forma comprobar sólo aquellos formatos de codificación que se ajusten a estas características. En el artículo (Bri y col. 2013) explicamos con detalle cómo se lleva a cabo la obtención de características de un dispositivo móvil. Una vez obtenidas las características del dispositivo móvil, se ajusta el formato de codificación a dichas características. Lo podemos hacer como hemos comentado mediante el método inductivo de predicción o también mediante el método heurístico.

Una vez determinado el formato de codificación que más se ajuste al entorno de red y a las características del dispositivo móvil, el servidor controlador de vídeo seleccionará el servidor de vídeo más apropiado tomando en cuenta los recursos disponibles y la carga de la granja de servidores y se iniciará la transmisión con el formato de codificación seleccionado. Durante la transmisión, si la estimación de QoE supera un umbral se llevarán a cabo reajustes en la QoS para mejorar la entrega de QoE. Si con ello no se mejora se llevará a cabo transcodificación mediante el método inductivo de estimación o con el método heurístico. El proceso descrito se puede observar en la figura 6.10. Con esta información también podríamos llevar a cabo la transcodificación mediante el método heurístico.

6.3.2.2 Algoritmo del cliente

En la parte del cliente, en primer lugar, la aplicación IPTV móvil envía una petición al controlador de vídeo para poder ver el canal solicitado. Cuando el servidor contesta con una aceptación, la aplicación envía al servidor controlador de vídeo características del dispositivo móvil como: la velocidad de procesamiento, la resolución de pantalla, etc. Tras recibir el controlador de vídeo dicha información, selecciona el servidor que más se ajuste a las características del dispositivo móvil. Seguidamente envía un enlace al dispositivo de usuario para que se conecte con dicho servidor y así poder iniciar la reproducción del contenido IPTV. Si no es posible se libera la conexión. La figura 6.11 muestra el algoritmo que acabamos de describir.

6.3.3 Protocolo de red

En este apartado describiremos el protocolo llevado a cabo entre el cliente móvil y el proveedor de servicios IPTV en un entorno de red móvil.

Cuando un nuevo usuario se une a la red envía un mensaje de petición (request message) de servicio IPTV al servidor controlador de vídeo con el fin de poder ver el contenido IPTV. El servidor controlador de vídeo estima la disponibilidad de

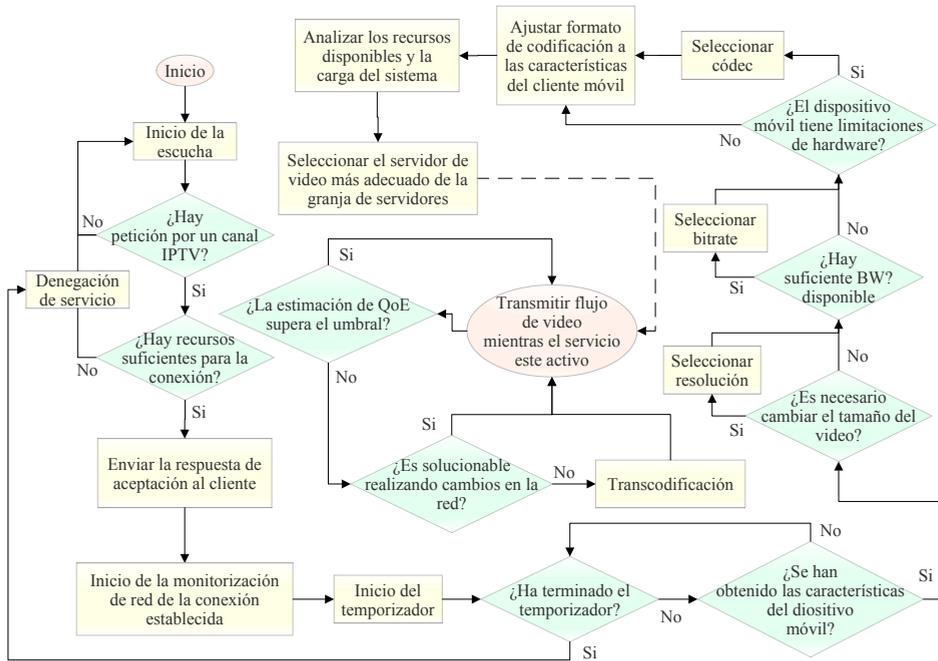


Figura 6.10: Algoritmo del controlador de vídeo para servicio de IPTV en redes móviles.

los recursos en la granja de servidores de vídeo que dependerá principalmente del número de peticiones recibidas y la información enviada periódicamente desde los servidores de vídeo al controlador de vídeo. Si hay suficientes recursos, el servidor controlador de vídeo envía un mensaje de aceptación (acceptance message) del servicio IPTV. Cuando el cliente recibe este mensaje, el dispositivo del usuario envía un mensaje respuesta con las características del dispositivo. Luego, el controlador de vídeo analiza dichas características y selecciona el formato de codificación que más se ajuste a ellas. El controlador de vídeo envía un mensaje de petición (request message) para esta conexión a la granja de servidores y cada servidor responde con su IP, su carga y su disponibilidad. Con esta información el controlador de vídeo decide cuál es la mejor opción y le envía la información del cliente. Entonces el servidor de vídeo contesta con un mensaje de reconocimiento (acknowledgment message) para la entrega de vídeo el cual además contiene el puerto UDP utilizado para esta conexión. El servidor controlador de vídeo envía la IP del servidor y el puerto con un mensaje del tipo: *rtsp://direccionIP:port/stream.sdp*. Este mensaje será utilizado por la aplicación (e.g. reproductor cliente VLC) del dispo-

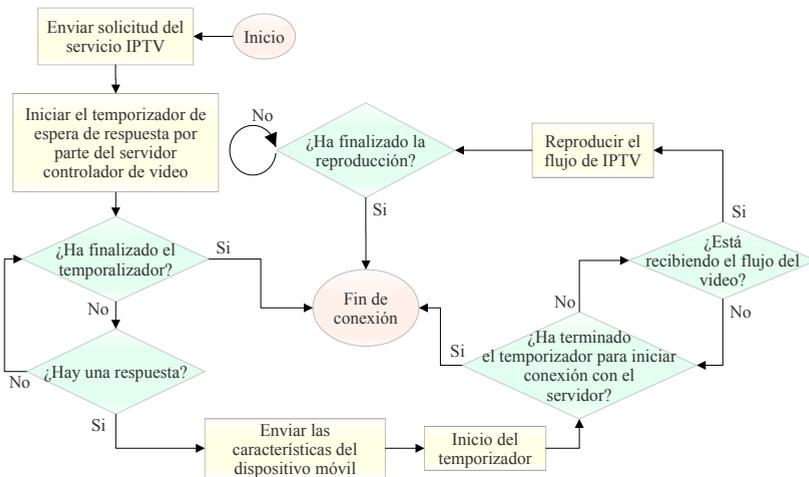


Figura 6.11: Algoritmo del cliente IPTV en redes móviles.

sitivo del usuario para iniciar el contenido IPTV. Finalmente, el dispositivo del usuario recibirá el flujo de vídeo. Este protocolo puede ser analizado en la figura 6.12.

Como hemos visto en esta sección, el algoritmo de gestión para entornos móviles que hemos planteado se basa en características de los dispositivos móviles y en los parámetros de red para la toma de decisiones. En los siguientes apartados analizaremos ambos aspectos.

6.3.4 Análisis de las características de los dispositivos móviles

El algoritmo de gestión toma decisiones basándose en características relacionadas con los dispositivos móviles como son: el consumo de ancho de banda, la tasa de compresión, el tiempo de transcodificación, la resolución de pantalla, la capacidad de procesamiento, etc.. En la subsección 4.2.3 hemos llevado a cabo un estudio en el que analizamos cada una de estas características en entornos reales de simulación. Los resultados obtenidos en dicho estudio son muy importantes para el desarrollo del sistema de gestión ya que permite seleccionar el vídeo más apropiado para cada dispositivo móvil según estas características. En la subsección 4.2.3 también se presenta una tabla en la que se muestra las principales características de los dispositivos móviles.

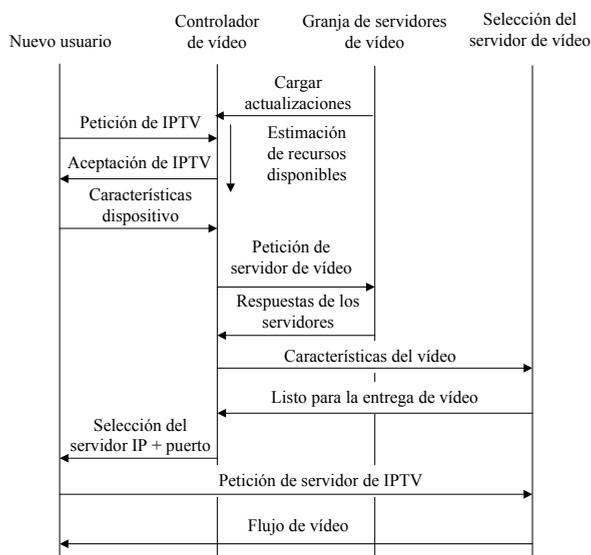


Figura 6.12: Protocolo de red en una transmisión móvil.

Cabe decir que cualquier proceso de transcodificación con un tiempo superior a un segundo no es factible para el algoritmo que hemos planteado, ya que produciría un retardo en el sistema que podría afectar a la QoE.

6.3.5 Análisis de los parámetros de red y sistemas de codificación en entornos móviles

En este apartado estudiaremos el impacto de los parámetros de red y de los sistemas de codificación en la QoS en transmisiones de flujos de vídeo en entornos de red móviles. Para ello, analizaremos parámetros de red como son: el jitter, el retardo y los paquetes perdidos en entornos de red UMTS y WIFI (IEEE 802.11g). Para este estudio codificamos un vídeo fuente en: H264, MPEG-4 y XviD. A continuación, encapsulamos cada uno de estos vídeos en: .mp4, .mkv, .avi, .3g2, .wmv, y .3gp. Las características del vídeo fuente son: duración del vídeo 80 seg, con una resolución de 800x480 pixels, una relación de aspecto de 16:9, bitrate de 384Kbps, a 15 fps, un bitrate de audio de 32 Kbps, muestreo de audio de 16 Kilohertz (KHz) y un tamaño total de 3.7 Mb.

En las gráficas de los resultados obtenidos en este estudio hemos considerado mostrar las 300 primeras muestras del vídeo recibido por el dispositivo móvil, que vienen a ser 20 seg., ya que al inicio de la transmisión es dónde se producen comportamientos más irregulares y con este tamaño se pueden ver con más detalle.

6.3.5.1 Transmisión en red UMTS

El primer parámetro que hemos analizado ha sido el **RETARDO**. En la figura 6.13 se puede observar varias transmisiones de flujo de vídeo codificados en H264, cada uno encapsulado en un contenedor distinto, de 20s de duración y a través de una red UMTS. Analizando esta figura vemos que para todas las transmisiones el retardo es muy similar salvo el vídeo con el contenedor .wmv que al inicio de la transmisión sufre un retardo bastante considerable con respecto al resto pero decrece rápidamente. También cabe destacar el aumento del retardo en la transmisión en un cierto intervalo de tiempo del contenedor .mkv. En conclusión, no podemos decir que haya un contenedor que destaque por su bajo retardo durante la transmisión aunque .mp4 parezca que sea ligeramente el que mejor resultado haya dado pese a no ser todo el tiempo.

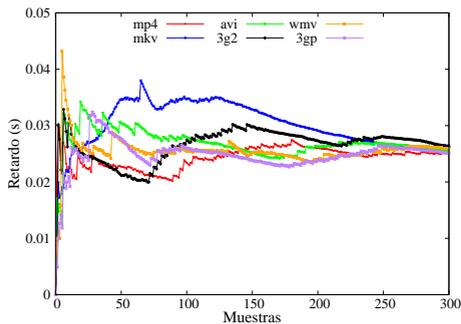


Figura 6.13: Retardo de H264 en UMTS.

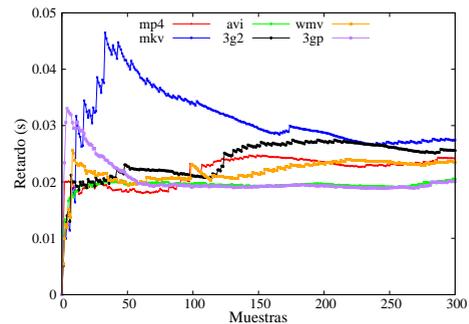


Figura 6.14: Retardo de MP4 en UMTS.

Cuando la codificación utilizada es MP4 el que peor resultado da en términos de retardo en la transmisión del vídeo es cuando se utiliza .mkv, tal como podemos observar en la figura 6.14. Vemos también, con este mismo contenedor, un pico de retardo alrededor de los 4s de transmisión. Al contrario que .mkv, los contenedores .avi y .3gp son los que mejor resultado dan con una media de retardo de 0.02s.

Cuando la codificación utilizada es XviD, el que peor comportamiento tiene en términos de retardo es .mp4, tal como podemos observar en la figura 6.15. Aunque el contenedor .avi tiene un pico al inicio de la transmisión después se estabiliza. El contenedor con menor retardo es .mkv.

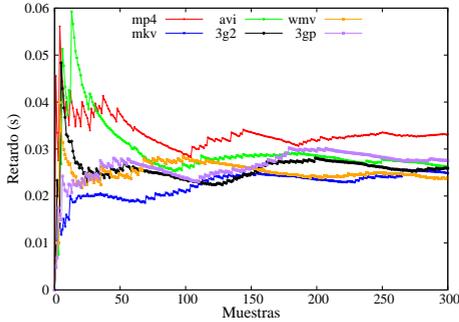


Figura 6.15: Retardo de XVID en UMTS.

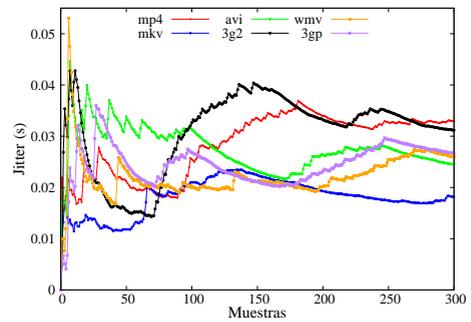


Figura 6.16: Jitter de H264 en UMTS.

A continuación se analizó el **JITTER**. En la figura 6.16 se puede observar varias transmisiones de flujo de vídeo codificados en H264, cada uno empaquetado en un contenedor distinto, de 20s de duración y a través una red UMTS. Analizando esta figura vemos que el peor contenedor en los primeros 8.3s de transmisión es .avi, aunque el pico más alto de jitter corresponde a .wmv. A partir de los 8.3s los contenedores que peor resultado dan son .mp4 y .3g2. El contenedor que mejor resultado da con respecto al jitter es .mkv.

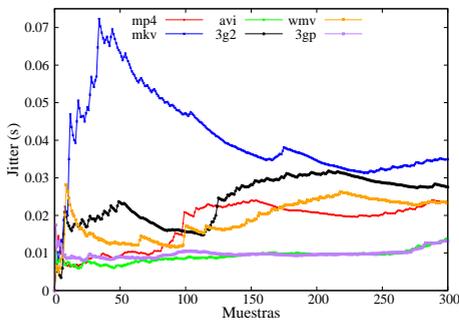


Figura 6.17: Jitter de MP4 en UMTS.

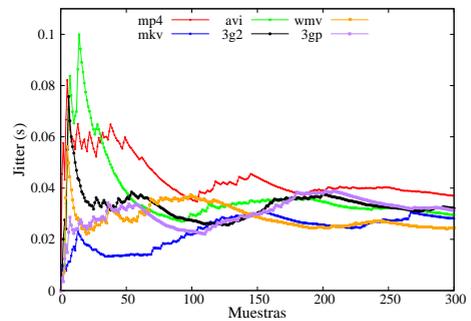


Figura 6.18: Jitter de XVID en UMTS.

Cuando la codificación utilizada es MP4, tal como podemos observar en la figura 6.17, el peor contenedor es .mkv. Tiene dos picos considerables de 0.07s y 0.03s. de jitter. Por el contrario los que dan mejor resultado en valores de jitter son los contenedores .avi y .3gp. Ambos tienen valores de jitter por debajo de los 0.01s.

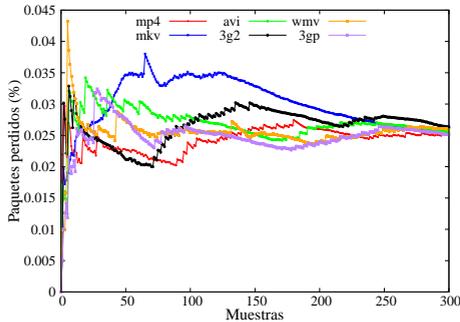


Figura 6.19: Pérdida de paquetes de H264 en UMTS.

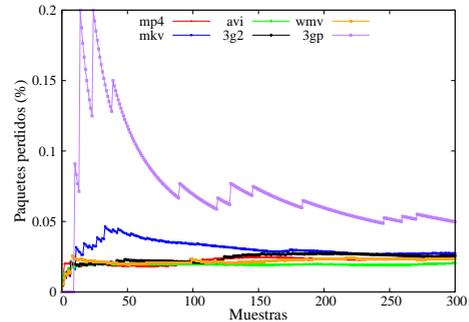


Figura 6.20: Pérdida de paquetes de MP4 en UMTS.

Cuando la codificación utilizada es XviD, se puede observar en la figura 6.18 que el contenedor que peor resultado de jitter da es .mp4. aunque el pico de mayor jitter corresponda al contenedor .avi. El contenedor de vídeo con un jitter medio más bajo es .mkv.

Finalmente se analizaron los **PAQUETES PERDIDOS**. En la figura 6.19 se puede observar varias transmisiones de flujo de vídeo codificados en H264, cada uno empaquetado en un contenedor distinto, de 20s de duración y a través una red UMTS. Analizando esta figura vemos que .wmv tiene el pico más alto en pérdidas de paquetes aunque después de este pico va igualándose al resto de contenedores. El contenedor con una mayor pérdida de paquetes durante la transmisión es .mkv.

Cuando la codificación utilizada fue MP4, tal como podemos observar en la figura 6.20, el que más pérdidas de paquetes ha sufrido durante la transmisión ha sido .mkv. Los que menos .avi y .3gp. La media de paquetes perdidos por .avi y .3gp es del 0.02 %.

Cuando la codificación utilizada es XviD, tal como podemos observar en la figura 6.21, el pico más alto en cuanto a pérdidas es con el contenedor .avi. Sin embargo, el contenedor con más pérdidas de paquetes es .mp4 con una media de 0.03 %. El que menos .mkv.

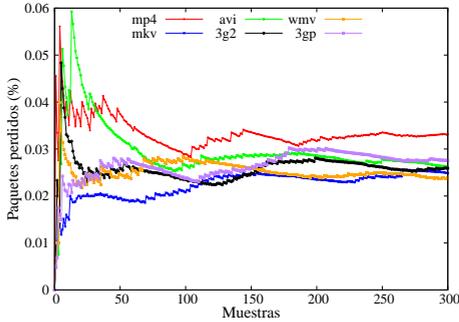


Figura 6.21: Pérdida de paquetes de XVID en UMTS.

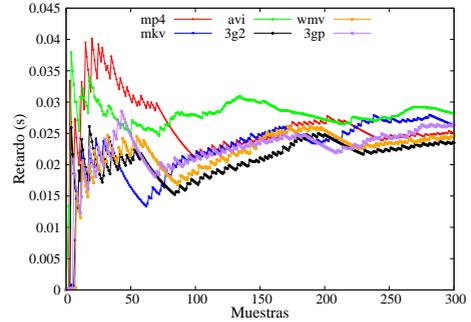


Figura 6.22: Retardo de H264 en WIFI.

6.3.5.2 Transmisión en red WIFI

El primer parámetro que hemos analizado en relación a este estudio es el **RE-TARDO**. En la figura 6.22 se puede observar varias transmisiones de flujo de vídeo codificados en H264, cada uno empaquetado en un contenedor distinto, de 20s de duración y a través una red WIFI. Analizando esta figura vemos que con el contenedor .avi el retardo es el más alto durante la transmisión del flujo de vídeo. Cabe destacar que entre los segundos 1.25 y 6.25 de la transmisión el contenedor con más retardo es .mp4. En este entorno de red el contenedor con menos retardo es .3g2. El contenedor .wmv también dio muy buenos resultados.

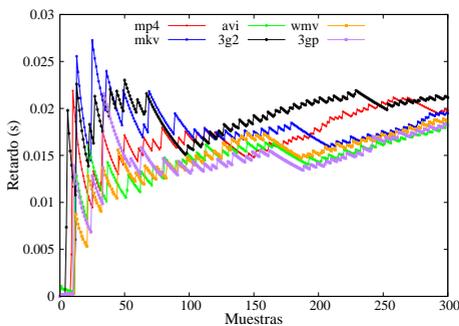


Figura 6.23: Retardo de MP4 en WIFI.

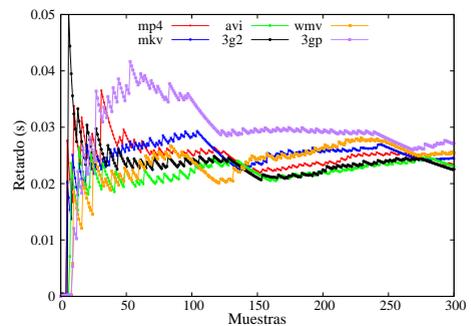


Figura 6.24: Retardo de XVID en WIFI.

Cuando la codificación utilizada es MP4 los contenedores que menos retardo sufrieron durante la transmisión fueron .3gp, .wmv y .avi tal como podemos observar en

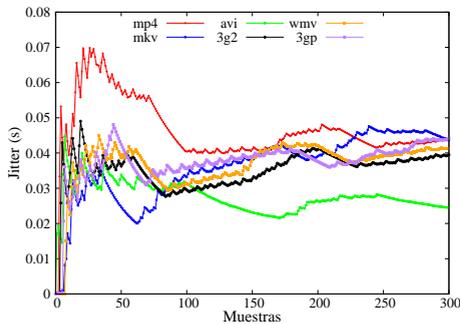


Figura 6.25: Jitter H264 en WIFI.

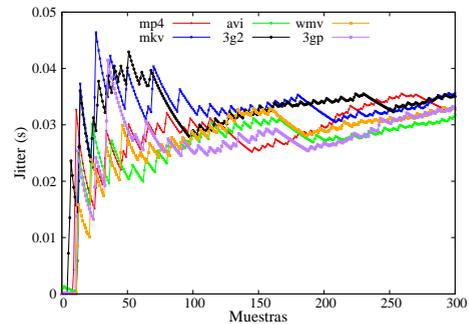


Figura 6.26: Jitter de MP4 en WIFI.

la figura 6.23. Podemos observar también que conforme transcurre la transmisión el retardo aumenta para todos los contenedores.

Cuando la codificación utilizada es XviD, tal como podemos observar en la figura 6.24, el contenedor con mayor retardo durante la transmisión de forma notable es .3gp. Los mejores resultados fueron para .avi y .3g2. Cabe destacar que llega un punto en el que todos los contenedores se estabilizan en términos de retardo. A partir de este momento los valores oscilan entre 0.02s y 0.03s

A continuación analizamos el **JITTER**. En la figura 6.25 se puede observar varias transmisiones de flujo de vídeo codificados en H264, cada uno empaquetado en un contenedor distinto, de 20s de duración y a través una red WIFI. El peor contenedor en términos de jitter es .mp4. Tiene varios picos cerca de los 0.07s. Pasados los 8,7s se observa que el mejor contenedor es .avi con valores entre 0.03s y 0.04s.

Cuando la codificación utilizada es MP4 se obtienen unos resultados similares para todos los contenedores tal como se puede observar en la figura 6.26. La mayor diferencia se encuentra en los 8.3 primeros segundos. Para este intervalo de tiempo los peores contenedores en términos de jitter son .mkv y .3g2, mientras que el mejor es .avi. Tras los 20s todos los contenedores tienen un jitter bastante estable con valores entre 0.03s y 0.035s

Cuando la codificación utilizada es XviD, el peor contenedor es .3gp y el mejor .avi, este último con valores por debajo de 0.04s. Estos resultados pueden ser comprobados en la figura 6.27.

Finalmente analizamos los **PAQUETES PERDIDOS**. En la figura 6.28 se puede observar varias transmisiones de flujo de vídeo codificados en H264, cada uno empaquetado en un contenedor distinto, de 20s de duración y a través una red

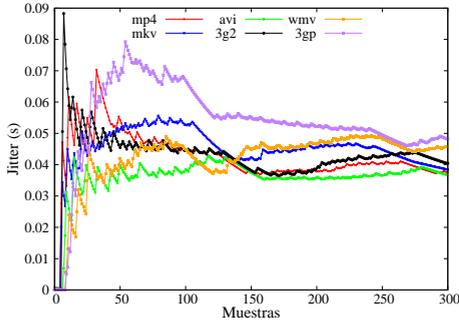


Figura 6.27: Jitter de XVID en WIFI.

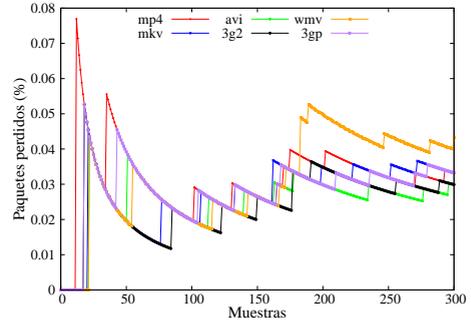


Figura 6.28: Pérdida de paquetes de H264 en WIFI.

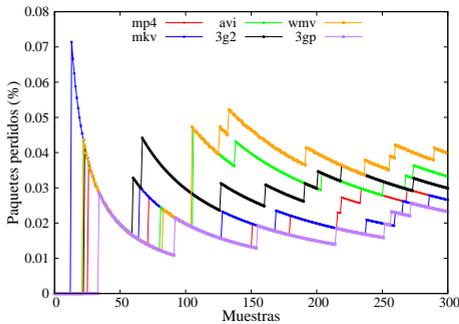


Figura 6.29: Pérdida de paquetes MP4 en WIFI.

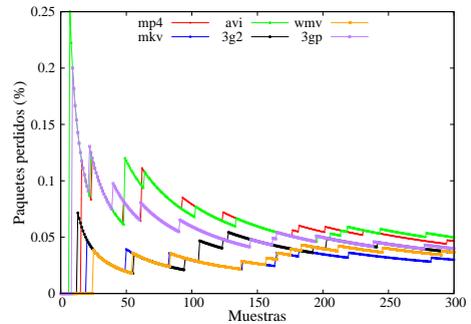


Figura 6.30: Pérdida de paquetes de XVID en WIFI.

WIFI. Analizando esta figura vemos que para todos los contenedores la pérdida de paquetes durante la transmisión es bastante similar, aunque después de los 15s podemos considerar como el peor contenedor en cuanto a pérdida de paquetes el .wmv.

Cuando la codificación utilizada es MP4 los contenedores con menores pérdidas son .mkv y .3gp, tal como podemos observar en la figura 6.29. El que más pérdidas tiene durante la transmisión es .wmv.

Cuando la codificación utilizada es XviD, se observa que después del segundo 14.5 las pérdidas de paquetes para todos los contenedores están alrededor del 0.05 % tal como podemos observar en la figura 6.30. Sin embargo, al inicio de la transmisión

los valores son bastante diferentes como por ejemplo el caso de .avi que tiene un pico de pérdidas de paquetes del 0.25 %.

Para ampliar la información sobre este estudio se puede consultar el trabajo (Lloret y col. 2012).

6.3.6 Conclusiones

En esta sección se ha presentado un sistema de gestión para redes ubicuas. Hacemos referencia a redes ubicuas ya que el usuario puede realizar la conexión desde cualquier lugar. El sistema de gestión se basa en las características del dispositivo móvil del usuario final y de las condiciones de red para estimar la QoE. A partir de esta información se selecciona el mejor servidor de vídeo de la granja de servidores con el que se conectará el cliente y así iniciar la transmisión de contenido IPTV en las mejores condiciones de QoE posibles.

Para llevar a cabo la comunicación cliente - servidor se ha desarrollado un protocolo de comunicación. También se ha analizado los parámetros de QoS con respecto a los formatos de codificación y a los contenedores de vídeo para dispositivos móviles para entornos de redes UMTS y WIFI. De este estudio se puede concluir que durante el inicio de las diferentes transmisiones se producen picos considerables de retardo, de jitter y de paquetes perdidos. Esto es debido a que al iniciar la transmisión la cantidad de información que se envía es máxima y se llevan a cabo procesos de sincronización. Por otro lado, se puede observar que hay contenedores que dan muy buenos resultados con un tipo de códec y sin embargo con otros no. Por ejemplo, .mkv da muy buenos resultados a nivel de retardo, jitter y paquetes perdidos para el códec Xvid en entornos de red UMTS. Sin embargo, este mismo contenedor funciona muy mal tanto a nivel de retardo, como de jitter, como de paquetes perdidos para el códec MPEG-4 en el mismo entorno de red. Otro aspecto interesante es cómo ciertos contenedores en un entorno de red dan resultados positivos y sin embargo en el otro no. Por ejemplo, en un entorno de red UMTS, con un códec H264 y el contenedor .avi, los resultados de jitter son considerablemente malos. Sin embargo, si el entorno es WIFI con el mismo formato de vídeo los resultados son positivos. Para finalizar, como las pérdidas de paquetes son un factor muy importante a la hora de garantizar la QoE del usuario final es importante que destaquemos aquellos contenedores que dan buenos resultados y los que no. Para H264 .wmv da malos resultados en ambos entornos de red. Para MPEG-4, .avi y .3gp funcionan bien para entornos UMTS y sin embargo .3gp funciona mal en entornos WIFI y el que da buenos resultados es .wmv. Para XviD, .avi da malos resultados para ambos entornos de red.

Por último, cabe decir que en la subsección 4.2.3 hemos analizado el rendimiento de los servidores en aspectos como: el consumo de ancho de banda, la tasa de compresión y el tiempo de transcodificación para diferentes dispositivos móviles.

6.4 Sistema de gestión basado en métodos inteligentes y su estudio en la transmisión de vídeo estereoscópico

6.4.1 Introducción

Uno de los aspectos importantes en una transmisión de flujo de vídeo es que no haya interrupciones en la señal con el fin de entregar una adecuada calidad. Evitar una parada durante la reproducción del vídeo es prácticamente más importante que el que haya una disminución en la calidad del vídeo recibido (Silverman y Hanrahan 2012). Cuando la transmisión de flujo de vídeo es 3D/ST este aspecto todavía se complica más, debido entre otros factores a los dispositivos específicos para la visualización de 3D/ST, ya que necesitan más información y una mayor calidad de imagen si se desea cumplir las expectativas del usuario.

Un parámetro de red que debemos analizar en profundidad en este tipo de transmisiones es el retardo ya que la QoE se podría ver afectada por nuevos factores como el cambio de vista 3D. En (Kim y col. 2007) analizan el retardo dado por el cambio de vista y cómo podrían afectar a la QoE. Con el fin de solucionar este problema tratan de reducir la diferencia de retardo de los nodos de origen y de destino.

El sistema de gestión que planteamos se basa en un sistema de gestión inteligente basado en estimaciones de la QoE. Como parámetros de entrada al sistema utilizaremos parámetros de QoS. En esta sección describiremos como hemos llevado a cabo el desarrollo de este sistema de gestión y del método inductivo de predicción que utiliza.

6.4.2 Algoritmo de gestión para la mejora en la entrega de vídeo 3D/estéreo

En esta sección planteamos un algoritmo de gestión (figura 6.31) con el que se pretende mejorar la entrega de vídeo estereoscópico para IPTV utilizando métodos inductivos de estimación basados en aprendizaje supervisado. Hemos realizado un desarrollo teórico de estos métodos en la sección 5.2. Para configurar el corpus de trabajo hemos realizado tareas de pre-procesamiento de los datos y etiquetado de las muestras. Cada muestra del corpus está formada por un vector de características y su correspondiente valor de QoE. Los valores medios del jitter, retardo, paquetes perdidos, ancho de banda y el formato de codificación conforman cada una de las características del vector de características. El valor asociado a cada muestra es el valor de la QoE extraído tanto de forma subjetiva como objetiva a partir del vídeo clip de referencia y el vídeo clip degradado recibido por el usuario. A partir de este corpus hemos realizado tareas de aprendizaje y test sobre los métodos de predicción con el fin de adaptarlos al problema, de tal forma que

dadas unas determinadas condiciones de red y un formato de codificación seremos capaces de predecir la QoE. Luego en esta sección vamos a desarrollar un método inductivo de predicción para usuarios de entornos cableados y dispositivos 3D/ST. Cabe decir que el procedimiento que vamos a explicar es extendible a cualquier otro entorno y características de usuario (móvil, PC, tablet, etc ...), no cambiaría en nada.

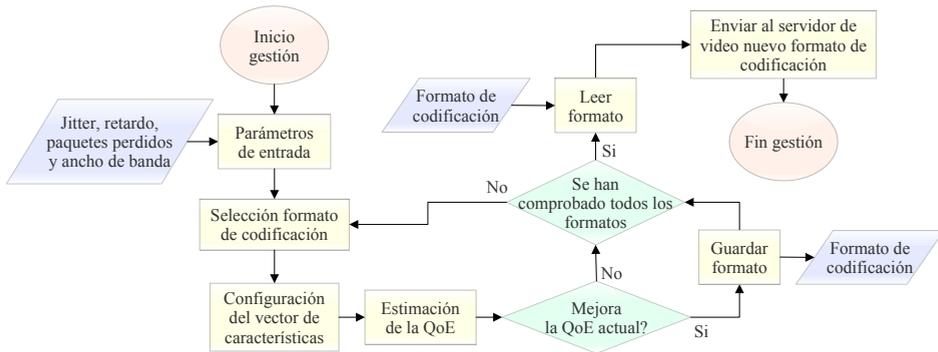


Figura 6.31: Algoritmo de gestión para la mejora en la entrega de vídeo 3D/ST .

Una vez definido el método de predicción iniciaremos la explicación del funcionamiento del algoritmo de gestión (figura 6.31). Durante la transmisión de un flujo de vídeo el controlador está a la escucha, en espera de recibir una alarma. Cuando se recibe una alarma y no es solucionable gestionando los parámetros de red, se inicia el proceso de gestión en base a la transcodificación. Para ello, en primer lugar el controlador de vídeo a partir de los parámetros de red: el jitter, el retardo, los paquetes perdidos y el ancho de banda, crea un vector de características que se utilizará como entrada del sistema de gestión. A partir de dicho vector el sistema de gestión inicia una búsqueda del formato de codificación con el que se mejore el valor actual de QoE. Para ello, se prueba cada uno de los formatos y se estima el valor de la QoE para cada uno de ellos. Los formatos de codificación que probaremos dependerán del tipo de cliente y del tipo de entorno. Por tanto, no probaremos todos los formatos de codificación posibles sino aquellos que se ajusten a la transmisión. Una vez probados todos los formatos, aquel que maximice el valor de la QoE se selecciona como formato de codificación para la transmisión. El formato seleccionado se envía al servidor de vídeo para que cambie el flujo de vídeo correspondiente al nuevo formato. El proceso de búsqueda del formato que maximice la QoE ha sido desarrollado de forma teórica en la sección 5.2.

En definitiva hemos desarrollado un sistema de gestión que estima qué formato de codificación de vídeo será capaz de optimizar la actual transmisión de flujo de vídeo IPTV. En esta sección mostraremos cómo se ha desarrollado el sistema de gestión inteligente para un entorno cableado y para una transmisión de vídeo 3D/ST. Cabe recordar que el modelo ha sido aplicado para este caso de estudio en concreto pero es extendible a cualquier entorno de red y usuario.

6.4.3 Pre-procesado de datos, extracción de características y etiquetado

El proceso para el desarrollo de nuestro sistema basado en métodos de inteligencia artificial consta de tres fases: pre-procesado de los datos, aprendizaje y test. En el siguiente apartado nos centraremos en el pre-procesado de los datos que hemos realizado para la configuración del corpus de trabajo.

6.4.3.1 Pre-procesado de datos

La topología utilizada para el desarrollo del banco de pruebas ha sido una configuración básica según el planteamiento descrito en la sección 3.9. Simulamos una red IPTV utilizando un servidor de vídeo VLC, un cliente VLC y un sistema evaluador basado en *Evalvid*. Los parámetros de red fueron ajustados utilizando un PC con Netdisturb. Este PC simula la red de distribución situada entre la cabecera de red y la red de acceso. El protocolo utilizado para la entrega del flujo de vídeo en tiempo real ha sido RTP. Para crear las secuencias de vídeo con formato 2D+Z y estéreo se utilizaron 3 secuencias de vídeo en crudo correspondientes a la vista izquierda, la vista derecha y a la de estimación de profundidad. Para extraer el formato 2D+Z se utilizaron la vista izquierda y la de profundidad y para la estereoscópica la derecha y la izquierda. El formato de estas secuencias es YUV 4:2:0 y se descargaron de la web *Mobile 3DTV Solideyesight*. La secuencia 2D+Z se comprimió utilizando H.264 Auxiliary Picture Syntax con valores de cuantización de Qs de 2, 4 y 8. La secuencia de vídeo estéreo la codificamos utilizando la codificación H.264/AVC MVC con valores de Qs de 2, 4 y 8. Las características comunes para ambos vídeos son: resolución de 1080x720 con escaneado progresivo de (720) a 25 fps. El programa utilizado para llevar a cabo el proceso de codificación es *JMVC* (Joint Multiview Video Coding (JMVC)). Para transmitir los vídeos se encapsularon con un formato contenedor .mp4. El programa utilizado para ello fue *FFMPEG*. Tras la codificación, se obtuvieron las siguientes tasas de bit: para 2D+Z un valor medio de 5282kb/s, 2438kb/s y 926kb/s para un Qs de 2, 4 y 8 respectivamente y para estéreo un valor medio de 9650kb/s, 4399kb/s y 1464kb/s para un Qs de 2, 4 y 8 respectivamente.

A continuación realizaremos una serie simulaciones con el fin de evaluar cada uno de los formatos de codificación en distintas condiciones de red durante una transmi-

sión de flujo de vídeo. Para el estudio simulamos una red con un retardo constante de 30ms bajo unas condiciones de jitter tanto uniforme como exponencial. Los valores de jitter uniforme varían conforme a la siguiente función.

$$F(x) = \begin{cases} \frac{1}{\beta-\alpha} & \alpha < x < \beta \\ 0 & x < \alpha \text{ o } x > \beta \end{cases} \quad (6.2)$$

Como se puede observar la función sólo toma un valor diferente de cero cuando x está entre α y β . Para el estudio, los valores seleccionados fueron $\alpha = 1$ y $\beta = 20$. El motivo es porque con dicho rango se evitaba que los resultados salieran demasiado dispersos. Los valores de jitter exponencial siguen una función de densidad de probabilidad conforme a la siguiente función.

$$F(x) = \begin{cases} \lambda e^{-\lambda x} & x > 0 \\ 0 & x < 0 \end{cases} \text{ o } x = 0 \quad (6.3)$$

Para este estudio el valor de λ utilizado es de $\lambda = 0,1$. También se provocaron pérdidas en la red con valores constantes del 0,1 % y 0,01 %. Este tipo de análisis está fundamentado en base a trabajos como (Calyam y col. 2004) y según la (ITU-T/G.114 2003). En la tabla 6.1 presentamos de forma esquemática cuales son las simulaciones que vamos a presentar en este estudio.

Tabla 6.1: Simulaciones para la evaluación de los distintos formatos de codificación y el MOS.

Flujo	Jitter	Retardo (ms)	Pérdidas (%)	Codificación	Q_s
1	uniforme	30	0.1	2D+Z	2
2	uniforme	30	0.1	estéreo	2
3	uniforme	30	0.1	2D+Z	4
4	uniforme	30	0.1	estéreo	4
5	exponencial	30	0.01	2D+Z	2
6	exponencial	30	0.01	estéreo	2
7	exponencial	30	0.01	2D+Z	4
8	exponencial	30	0.01	estéreo	4

Para cada una de estas transmisiones calculamos también el PSNR y el MIV comparando el vídeo antes de cada transmisión y después. El motivo es que utilizaremos

estas medidas como apoyo a la valoración subjetiva de la calidad de vídeo con la que realizaremos el etiquetado de las muestras. Antes de analizar en profundidad cuál ha sido la respuesta de los diferentes formatos de codificación a las distintas condiciones de red, vamos a comprobar cuál es la calidad objetiva basada en la VQM de cada uno de los flujos de vídeos y comparar así los diferentes formatos de codificación.

Un aspecto muy interesante es la calidad de vídeo que presenta cada uno de los formatos codificados de vídeo. Para ello, procedimos a analizar la VQM de cada uno de ellos. En (Pinson y Wolf 2004) proponen VQM como una medida objetiva para percibir la calidad de vídeo 2D. Con esta medida se percibe deficiencias en el vídeo como: la visión borrosa, movimiento poco natural, ruido global, la distorsión de bloque y distorsión de color. El resultado es una combinación de todas ellas en una única métrica. Según (Hewage y col. 2009), la métrica de evaluación objetiva de vídeo VQM se puede utilizar para evaluar QoE 3D estereoscópico. En base a este criterio llevaremos a cabo la evaluación de la calidad de vídeo de los diferentes formatos de codificación que utilizaremos en este estudio, siendo conscientes de que no tienen una alta correlación tal como se ha demostrado en (Joveluro y col. 2010). Para el cálculo de VQM se ha utilizado el programa (*Elecard Video Quality Estimator*), tal como se explica en (Atayero y col. 2011). Para ello se pasa al programa como parámetros de entrada el vídeo original y el procesado. En la figura 6.32 se muestran los resultados obtenidos de VQM para varios formatos de codificación de vídeo estereoscópico, que al igual que el resto de experimentos, durante los 2 primeros minutos de la transmisión.

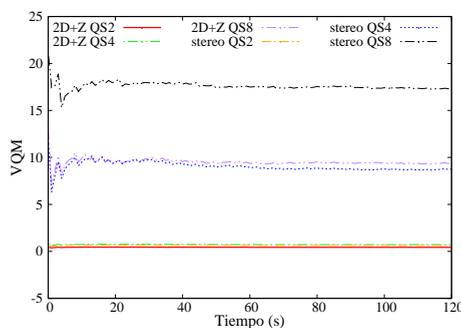


Figura 6.32: Comparación de la VQM de diferentes flujos de vídeo con distintos formatos de codificación.

La figura 6.32 muestra la VQM de diferentes formatos de codificación. Las escalas de cuantificación utilizadas han sido 2, 4 y 8 y los formatos de codificación estereoscópica utilizados han sido 2D+Z y estéreo. Cuanto más próximo es el valor

de VQM a cero mejor es la calidad de vídeo. En base a esto, observamos que los vídeos con un $Q_s=2$ tienen muy buena calidad. Por el contrario, los vídeos con un $Q_s=8$ tienen una calidad bastante inferior. Un flujo con $Q_s=4$ y un formato de codificación 2D+Z también da muy buenos resultados en valores de VQM. Cuando el formato es estereoscópico, la VQM obtenida ya no es tan buena, pero como el ancho de banda necesario para su transmisión es inferior al de $Q_s=2$ nos será muy útil para el estudio. De forma global, 2D+Z da mejores resultados que estéreo en valores de VQM. Para analizar con más profundidad cómo responden los diferentes formatos de codificación a las distintas condiciones de red, mostraremos el análisis de cada una de las características llevado a cabo durante la transmisión de los distintos flujos de vídeo. En el resto de experimentos analizaremos los resultados obtenidos con $Q_s=2$ frente a $Q_s=4$ ya que la finalidad es describir el proceso para la configuración del corpus. Por tanto, debemos tener en cuenta que los resultados de los experimentos realizados con $Q_s=8$ también serán incluidos en el corpus además de otras de experimentos con otras condiciones de red que no hemos mostrado en este estudio.

En el siguiente experimento analizamos el **JITTER**. Para ello estudiamos el comportamiento de los formatos de vídeo 2D+Z y estereoscópico cuantificados a $Q_s=2$ y $Q_s=4$ cuando existe en la red jitter uniforme o exponencial. Los resultados obtenidos se pueden observar en la figura 6.33 y la figura 6.34.

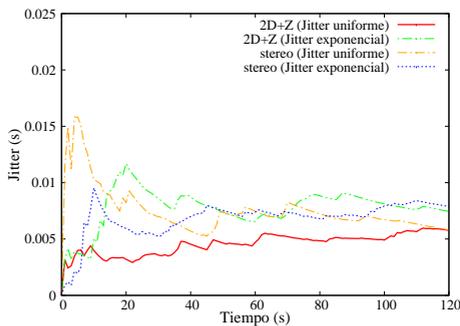


Figura 6.33: Jitter en la red del cliente durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.

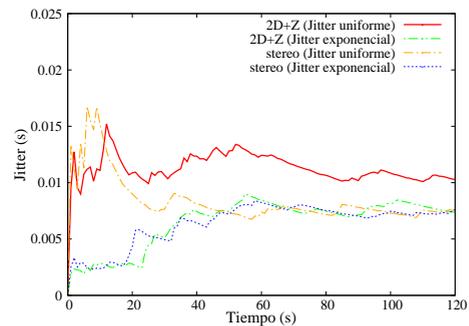


Figura 6.34: Jitter en la red del cliente durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.

Observando ambas figuras apreciamos que con $Q_s=4$, cuando el jitter es exponencial y al inicio de la transmisión se obtiene mejor respuesta, indistintamente del formato de codificación. Sin embargo, a partir de la mitad de la transmisión con $Q_s=4$ y $Q_s=2$ en cualquiera de los dos formatos de codificación se obtiene igual respuesta, aunque por calidad de imagen el sistema seleccionaría $Q_s=2$. Cuando

existe un jitter de red uniforme, clarísimamente un vídeo con un $Q_s = 2$ y con un formato 2D+Z respondería mejor a estas condiciones de red.

En el siguiente experimento analizamos el **RETARDO**. Para ello estudiamos el comportamiento de los formatos de vídeo 2D+Z y estereoscópico cuantificados a $Q_s = 2$ y $Q_s = 4$ cuando existe en la red un retardo constante bajo condiciones de jitter uniforme o exponencial. Los resultados obtenidos se pueden observar en la figura 6.35 y figura 6.36.

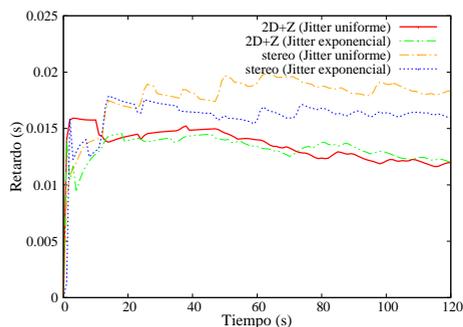


Figura 6.35: Retardo en la red del cliente durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.

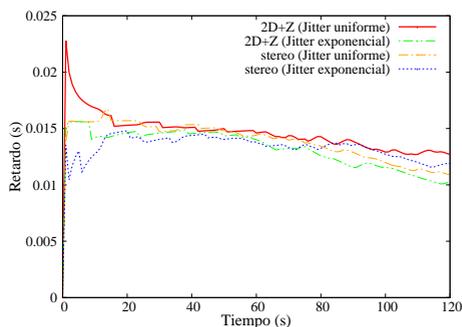


Figura 6.36: Retardo en la red del cliente durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.

Analizando ambas figuras, observamos que a medida que avanza la transmisión ambos sistemas de codificación van respondiendo mejor al retardo tanto si existe un jitter uniforme como exponencial, salvo con $Q_s = 2$ cuando el formato de codificación es estéreo especialmente cuando existe un jitter uniforme en la red. En este caso el cliente se ve afectado con un retardo creciente. Llegado un momento avanzado de la transmisión sí que parece que comience a responder mejor, pero sin llegar a obtener buenos resultados con respecto al resto. La respuesta del flujo 2D+Z para este mismo caso es mucho mejor, independientemente del jitter ya sea uniforme o exponencial. Si la transmisión es con un flujo de vídeo con $Q_s = 4$, tanto en formato 2D+Z como estéreo, la respuesta es muy similar, salvo al inicio de la transmisión en la que el flujo 2D+Z al haber un jitter uniforme en la red se ve afectado considerablemente.

En el siguiente experimento analizamos la **PÉRDIDA DE PAQUETES**. Para ello estudiamos el comportamiento de los formatos de vídeo 2D+Z y estereoscópico cuantificados a $Q_s = 2$ y $Q_s = 4$ cuando existe en la red pérdidas de paquetes bajo condiciones de jitter uniforme o exponencial. Las pérdidas provocadas en la red

han sido del 0.1% y el 0.01%. Los resultados obtenidos se pueden observar en la figura 6.37 y la figura 6.38.

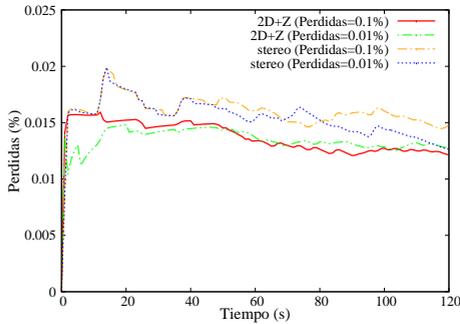


Figura 6.37: Paquetes perdidos en la red del cliente durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.

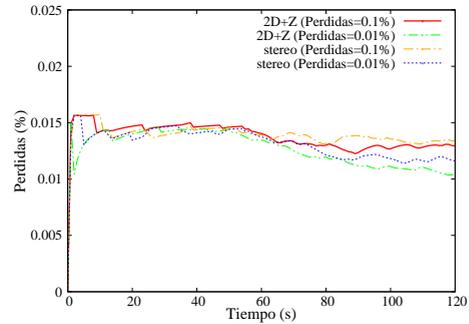


Figura 6.38: Paquetes perdidos en la red del cliente durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.

Observando la figura 6.38 vemos que tanto el flujo 2D+Z como el estéreo tienen una respuesta muy similar a la pérdida de paquetes en la red. Podríamos decir que la transmisión de un flujo 2D+Z provoca ligeramente menos pérdidas que el estéreo. Sin embargo, cuando la transmisión se lleva a cabo con un flujo con una cuantificación de vídeo de $Q_s = 2$ la situación cambia. Debemos destacar las pocas pérdidas de paquetes que se dan con una transmisión de flujo 2D+Z. El resultado es tan bueno que incluso en un contexto más desfavorable con pérdidas de paquetes en la red del 0.1%, mejora los resultados obtenidos con pérdidas del 0.01% en una codificación de $Q_s = 2$.

En el siguiente experimento analizamos el **ANCHO DE BANDA**. Para ello comprobamos el BW del vídeo recibido para los formatos de vídeo 2D+Z y estereoscópico cuantificados a $Q_s = 2$ y $Q_s = 4$ cuando existe en la red un jitter uniforme o exponencial. Los resultados obtenidos se pueden observar en la figura 6.39 y figura 6.40.

Comparando ambas figuras rápidamente apreciamos que el BW necesario para transmitir los flujos de vídeo con una cuantificación de $Q_s = 2$ es mucho mayor que el necesario para $Q_s = 4$. Del mismo modo, el BW para transmitir vídeo estereoscópico es mayor que el utilizado con formatos de codificación 2D+Z. Podemos apreciar también los picos de BW correspondientes a las transmisiones de imágenes I. Vemos que las imágenes I en 2D+Z transmiten menos información que en estéreo. Sin embargo, cuando la transmisión es con un formato estéreo en condiciones de jitter uniforme la periodicidad de dichas imágenes tienen un comportamiento bastante irregular.

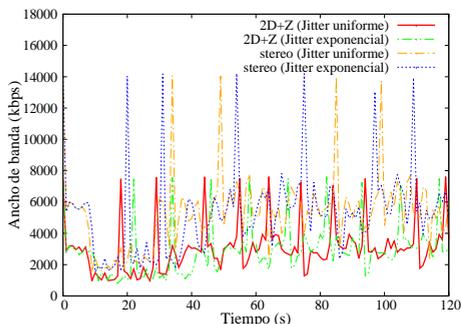


Figura 6.39: Ancho de banda del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS2 con formatos $2D + Z$ y vídeo estéreo.

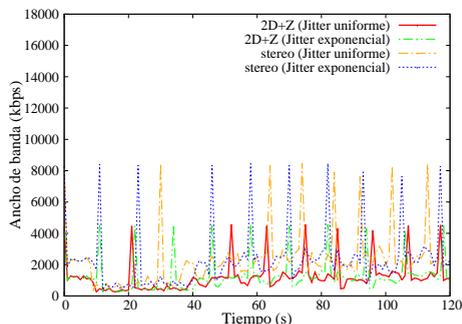


Figura 6.40: Ancho de banda del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS4 con formatos $2D + Z$ y vídeo estéreo.

En el siguiente experimento analizamos el **PSNR**. Para ello estudiamos la relación señal a ruido del vídeo recibido para los formatos de vídeo $2D + Z$ y estereoscópico cuantificados a $Q_s = 2$ y $Q_s = 4$ cuando existe en la red un jitter uniforme o exponencial. Los resultados obtenidos se pueden observar en la figura 6.41 y figura 6.42.

Si observamos ambas figuras podemos ver que a partir de cierto momento de la transmisión, el PSNR en el cliente sufre cierta estabilidad. Sin embargo, al inicio de la transmisión el PSNR es muy alto para todos los casos y disminuye de forma abrupta en poco tiempo. En este periodo de tiempo cabe destacar que dependiendo del jitter que exista en la red el PSNR en el cliente será más bajo o más alto. Podemos apreciar que no siempre el PSNR obtenido con $2D + Z$ es mayor que con estéreo, ya que con $Q_s = 4$ existe un periodo de tiempo, entre los 20 y 40 segundos, en el que el PSNR si el flujo de vídeo transmitido es estéreo con un jitter exponencial en la red supera al recibido con un flujo $2D + Z$ y jitter uniforme.

En este experimento se ha analizado el **MIV** vídeo recibido para los formatos de vídeo $2D + Z$ y estereoscópico cuantificados a $Q_s = 2$ y $Q_s = 4$ cuando existe en la red un jitter uniforme o exponencial. Los resultados obtenidos se pueden observar en la figura 6.43 y figura 6.44.

Analizando ambas figuras vemos que el MIV es considerablemente alto con respecto al resto cuando existe un jitter uniforme en la red y el flujo de vídeo transmitido es estéreo. Por el contrario, si el jitter que hay en la red es exponencial y el flujo de vídeo transmitido es $2D + Z$, el MIV obtenido en el cliente es más bajo que el resto. Esto ocurre tanto para $Q_s = 2$ como $Q_s = 4$. Para el resto de casos dependerá de

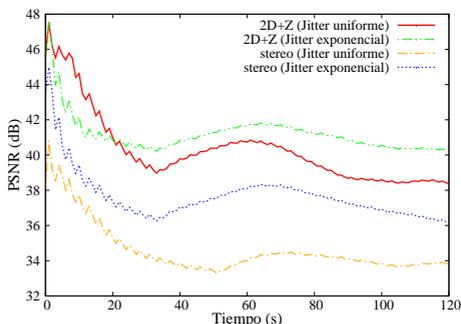


Figura 6.41: PSNR del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS2 con formatos 2D+Z y vídeo estéreo.

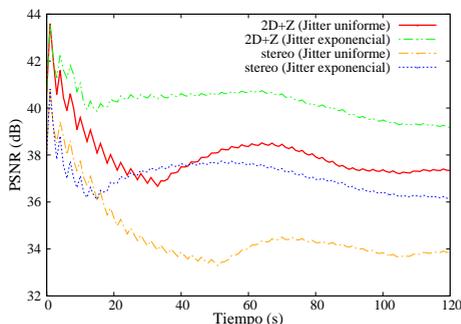


Figura 6.42: PSNR del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS4 con formatos 2D+Z y vídeo estéreo.

si la codificación es $Q_s = 2$ o $Q_s = 4$, el formato de vídeo y el tipo de jitter para que de un MIV más alto o más bajo.

6.4.3.2 Extracción de características

Para configurar cada uno de los vectores de características cogeremos todos los datos recibidos de QoS en intervalos de tiempo de 4s. Es decir, por ejemplo, tenemos el retardo de red durante una transmisión de flujo de vídeo con un formato 2D+Z y un $Q_s=2$ mostrado en la figura 6.35. Estos datos los dividiremos en intervalos de 4s. A continuación, por cada intervalo sacamos un valor medio y lo guardamos en el vector de características. Este proceso lo repetiremos para cada una de las características correspondiente a ese intervalo y a ese formato. Por cada vector tenemos que añadir también el formato de codificación del flujo de vídeo con el que se ha realizado la transmisión. Según el ejemplo, 2D+Z con un $Q_s=2$. Cada uno de los formatos ha sido enumerado con un valor entero, por lo que añadiríamos al vector de características el valor entero correspondiente a este formato. Finalmente obtendremos un vector de características por cada vídeo clip de 4s y por cada formato de codificación en diferentes condiciones de red. El siguiente paso será etiquetar cada una de las muestras.

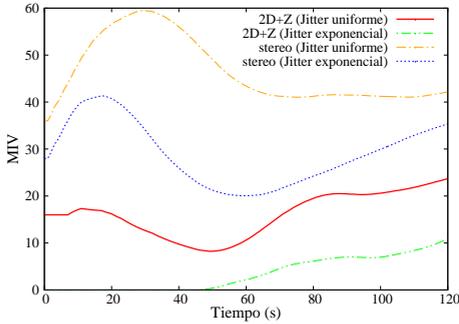


Figura 6.43: MIV del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS2 con formatos $2D + Z$ y vídeo estéreo.

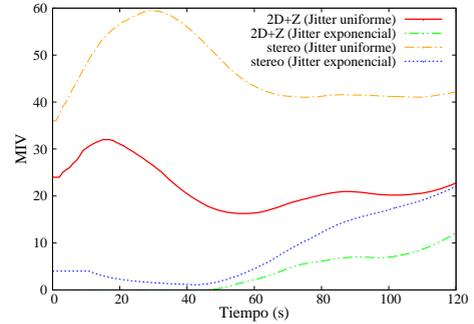


Figura 6.44: MIV del flujo de vídeo recibido durante una transmisión de flujo de vídeo codificado a QS4 con formatos $2D + Z$ y vídeo estéreo.

6.4.3.3 Etiquetado de muestras

Para etiquetar cada una de las muestras analizamos en primer lugar de forma subjetiva cada uno de los vídeo clips. Para llevar a cabo la evaluación, se seleccionaron 5 personas expertas en el procesamiento de imágenes. El método que utilizamos en la evaluación es el DSCQS. En este método, cada experto compara cada vídeo clip origen con el vídeo clip recibido y le da una puntuación de MOS. Los parámetros de percepción recomendables en la evaluación subjetiva de vídeos 3D/ST son: la calidad de imagen, la nitidez, la naturalidad y la percepción de profundidad. Para la percepción de la calidad de imagen se valoraron los mismos aspectos que para 2D como son: la falta de definición, el efecto fantasma y la pixelación, entre otros. Todos estos parámetros se han tratado en profundidad en la sección 3.4. Además de esta evaluación realizamos en segundo lugar una evaluación objetiva extrayendo el PSNR y el MIV de cada vídeo clip. Una vez comparado el flujo de vídeo origen con el recibido de forma objetiva y subjetivamente los etiquetamos usando un promedio entre ambas medidas. Es decir, promediamos las medidas de PSNR y MIV con el MOS para obtener una estimación de la QoE. Para ello reescalamos tanto la medida de PSNR como la del MIV de 0 a 5 (tal como se ha explicado en la sección 3.2). Tras el reescalado, promediamos ambas medidas con el MOS dando un peso del 60 % a las medidas objetivas y de un 40 % a las subjetivas. Tras el etiquetado dispondremos de un conjunto de muestras que utilizaremos para el entrenamiento de los métodos inductivos de estimación de forma supervisada.

6.4.4 Aprendizaje y test de los sistemas de predicción

Una vez obtenido el corpus de trabajo lo dividiremos en muestras de entrenamiento y en muestras de test. Con las muestras de entrenamiento se llevó a cabo el proceso de aprendizaje de los métodos de estimación con el fin de crear un modelo que se ajuste al problema. A continuación explicaremos el proceso de aprendizaje para los algoritmos evolutivos basados en RNRUP y MSVR. Para el proceso de experimentación hemos utilizado el software *KEEL* (Knowledge Extraction based on Evolutionary Learning) el cual es de código abierto basado en java, el *SVMlight* para experimentar de forma más concreta con las máquinas de soporte vectorial y el *JavaNNS* para experimentar con las redes neuronales.

6.4.4.1 Aprendizaje y test con el algoritmo evolutivo basado en RNRUP

El conjunto de datos utilizados en la experimentación en RNRUP consta de 275 muestras, de las cuales se han tomado 250 para entrenamiento (90 %) y 25 de test (10 %). Los parámetros de experimentación se han detallado en profundidad en subsección 5.2.2.2. Los principales parámetros de configuración con los que hemos trabajado han sido el número de nodos de la capa oculta y las funciones de activación para cada nodo de la capa oculta. El número de nodos de la capa oculta que hemos analizado han sido valores entre 1 y 15. Las funciones de activación analizadas han sido: Unidad Producto y Sigmoidal. Como técnica de validación hemos utilizado validación cruzada 10-Fold y 5-Fold y también un criterio de parada de máximo 1000 iteraciones. El mejor resultado obtenido ha sido con 4 nodos en la capa oculta y funciones de activación Unidad Producto con 10-Fold. Para determinar cuál ha sido el mejor resultado nos hemos basado en el error cuadrático medio (Mean Squared Error (MSE)). Los resultados del proceso de entrenamiento y test obtenidos en dicho experimento se muestran en la figura 6.45 y la figura 6.46.

6.4.4.2 Aprendizaje y test con el algoritmo evolutivo basado en MSVR

El conjunto de datos utilizadas en la experimentación en MSVR consta de 275 muestras, de las cuales se han tomado 250 para entrenamiento (90 %) y 25 de test (10 %). Los parámetros de experimentación se han detallado en profundidad en subsección 5.2.1.2. Los principales parámetros de configuración con los que hemos trabajado han sido:

- Factor de ponderación C . Hemos analizado valores entre 1 y 150 para este parámetro.

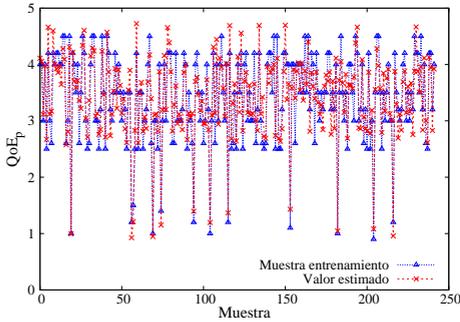


Figura 6.45: Proceso de entrenamiento para la RNRUP.

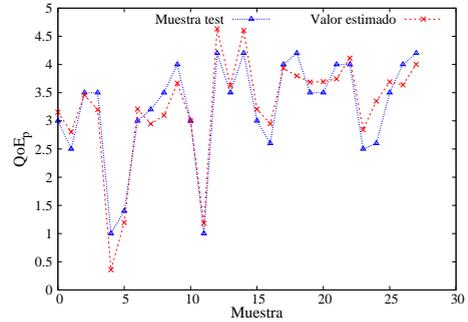


Figura 6.46: Proceso de test para la RNRUP.

- Factor de optimización ϵ . Hemos analizado valores entre 0.0001 y 1 para este parámetro.
- Función Kernel. Las dos funciones Kernel que hemos analizado han sido:
 - Polinomial: El principal parámetro con el que hemos experimentado en la función kernel Polinomial es con el valor d correspondiente al grado del polinomio. Hemos experimentado con valores desde 1 hasta 8 de este parámetro.
 - RBF: El principal parámetro con el que hemos experimentado en la función kernel RBF es con el valor γ . Hemos experimentado con valores entre 0.0001 hasta 1.

Como técnica de validación hemos utilizado validación cruzada 10-Fold y 5-Fold y también un criterio de parada de máximo 1000 iteraciones. El mejor resultado obtenido ha sido para el Kernel Polinomial: $C = 10$, $\epsilon = 0,1$ y $d = 3$, y para el Kernel RBF: $C = 10$, $\epsilon = 0,001$ y $\gamma = 1$. Para determinar cuál ha sido el mejor resultado nos hemos basado en el MSE. Los resultados del proceso de entrenamiento y test obtenidos para ambas funciones Kernel se muestran en la figura 6.47, la figura 6.48, la figura 6.49 y la figura 6.50.

Para evaluar qué modelo de predicción da mejores resultados analizaremos el MSE obtenido con cada uno.

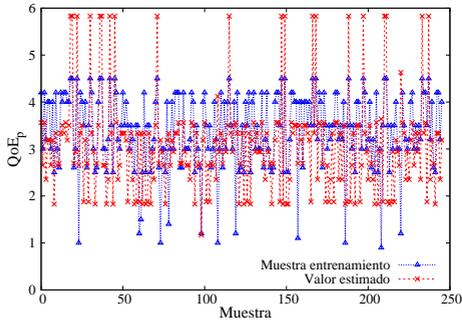


Figura 6.47: Proceso de entrenamiento para la MSVR con un Kernel Polinomial.

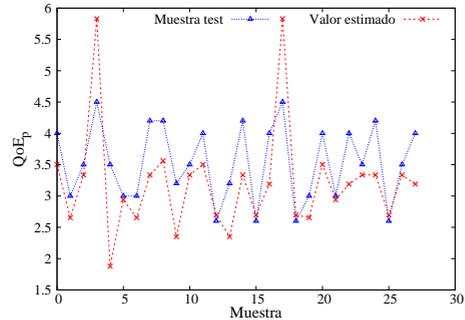


Figura 6.48: Proceso de test para la MSVR con un Kernel Polinomial.

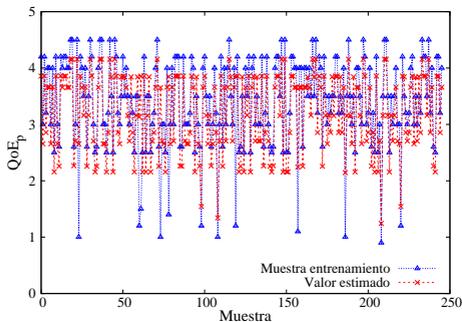


Figura 6.49: Proceso de entrenamiento para la MSVR con un Kernel RBF.

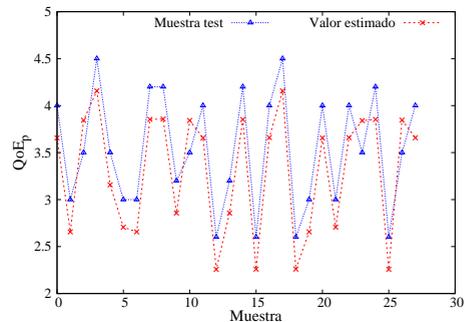


Figura 6.50: Proceso de test para la MSVR con un Kernel RBF.

6.4.4.3 Análisis de resultados basado en el MSE

En este apartado analizaremos estadísticamente los resultados obtenidos con los algoritmos RNRUP y MSVR. En el proceso de validación se compara la salida del modelo con la salida deseada, y se calcula el MSE del modelo. Este proceso se realiza tanto para las muestras de entrenamiento como para las de test. Con el MSE evaluamos la calidad de estimación, es decir cuanto se aproximan las estimaciones a los valores reales. En la figura 6.51 y en la figura 6.52 se muestra el MSE de cada estimador por cada una de las iteraciones realizadas con validación cruzada.

En función de todos los resultados obtenidos en el análisis anterior calcularemos el error medio de los diferentes MSE y la desviación estándar (Standard Deviation

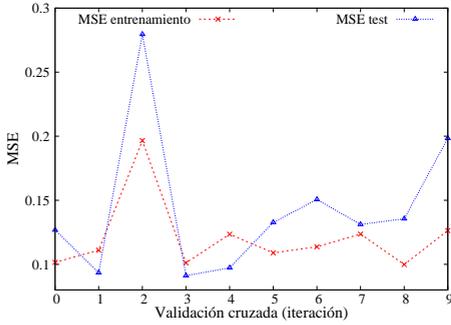


Figura 6.51: MSE de cada una de las iteraciones realizadas con validación cruzada para RNRUP.

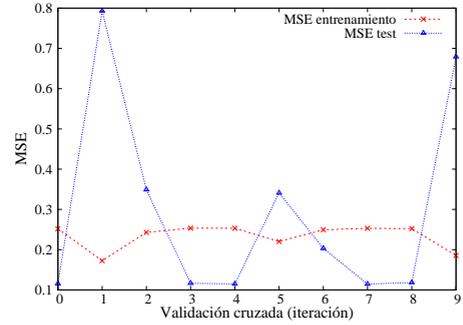


Figura 6.52: MSE de cada una de las iteraciones realizadas con validación cruzada para MSVR con kernel RBF.

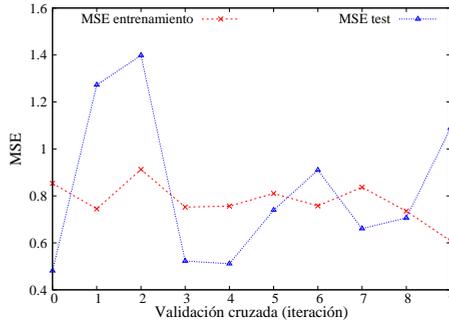


Figura 6.53: MSE de cada una de las iteraciones realizadas con validación cruzada para MSVR con kernel Polinomial

(STD)) de estos mismos valores. Los resultados obtenidos se muestran en la tabla 6.2.

Analizando los resultados de la tabla 6.2 vemos que el método de redes neuronales es el que mejor resultados da. Por el contrario la máquina de soporte vectorial con un kernel Polinomial no funciona nada bien, por lo que lo descartaremos para el siguiente estudio. La máquina de soporte vectorial con un kernel RBF se encuentra en una situación intermedia.

Tabla 6.2: Comparativa entre RNRUP y MSVR a nivel de MSE y STD tanto para entrenamiento como para test

Algoritmo	MSE TR	STD TR	MSE TS	STD TS
RNRUP	0.1206	0.0269	0.1436	0.0543
MSVR (RBF)	0.2334	0.0289	0.2945	0.2386
MSVR (POLI)	0.8290	0.3101	0.7768	0.0784

6.4.4.4 Test del sistema de gestión utilizando los métodos de predicción

Para finalizar este estudio llevaremos a cabo un test a fin de comprobar la precisión de los métodos de estimación, seleccionados en el estudio de la subsección anterior, al aplicarlos en el sistema de gestión. Para ello, realizaremos 7 transmisiones de vídeo con cada uno de los formatos de codificación por transmisión y bajo diferentes condiciones de red, las cuales no han sido incluidas en el corpus. El vídeo utilizado es el mismo que el utilizado en la subsección 6.4.3. Las características de red de cada test realizado se muestran en la tabla 6.3

Tabla 6.3: Descripción de las condiciones de red por cada test realizado.

Test	Condiciones de red
1	Una transmisión con pérdidas en la red del 0.8%
2	Una transmisión con un jitter uniforme en la red de 150ms
3	Una transmisión con un retardo uniforme en la red de 200ms
4	Una transmisión con un jitter uniforme en la red de 250ms
5	Una transmisión con un BW de 2Mb que provoca pérdidas de más del 1%
6	Una transmisión con un jitter exponencial y un retardo constante en la red de 200ms
7	Una transmisión con un jitter exponencial y un retardo constante en la red de 300ms

A continuación pedimos a 4 expertos que evalúen en valores de MOS cada uno de los vídeos recibidos y extraemos un valor medio. Este valor lo promediamos con el valor de PSNR y MIV del mismo vídeo recibido. Los resultados obtenidos se pueden observar en la figura 6.54, donde cada barra equivale a la estimación de la QoE promediada por cada uno de los formatos de codificación de los diferentes tests.

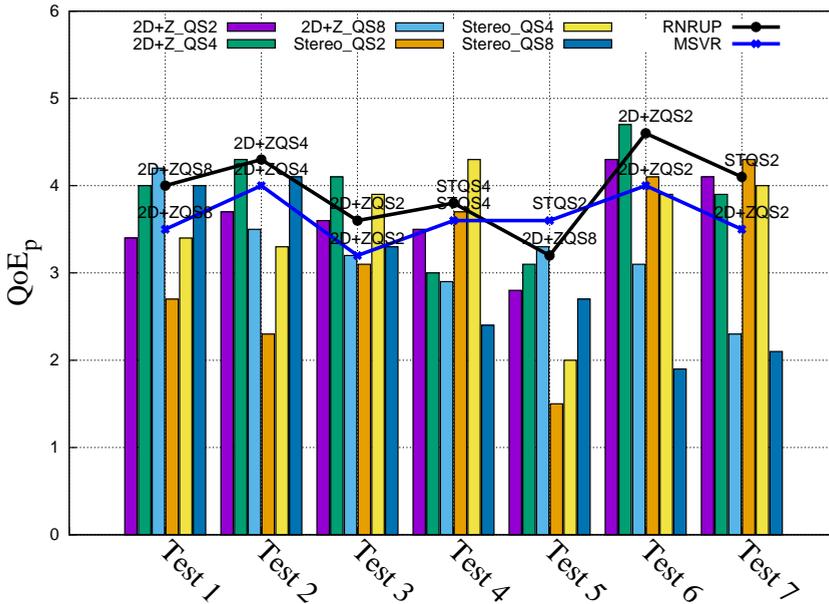


Figura 6.54: Resultados de los diferentes test realizados para la evaluación del sistema de gestión en un caso práctico.

Seguidamente, aplicaremos el sistema de gestión. Por cada test obtendremos los valores de jitter, retardo, paquetes perdidos y BW. Estos valores son añadidos como parámetros de entrada a cada método de estimación. A continuación, probaremos cada uno de los formatos de codificación junto con los parámetros de entrada y obtendremos una estimación de la QoE por cada formato. Aquel formato que obtenga un mayor valor de QoE es elegido por el sistema de gestión como formato óptimo para la transmisión. En la figura 6.54 podemos observar los resultados obtenidos por el sistema de gestión con cada método de estimación MSVR y RNRUP.

Vemos claramente que el método basado en redes neuronales da mejores resultados de aproximación que el basado en máquinas de soporte vectorial. Tiene un alto índice de aproximación en cuanto a la selección del códec y a la estimación de la QoE. En concreto, en la elección del formato de codificación, el método de RNRUP en el test 6 falla en la elección y en el resto de los casos acierta. En cambio, el método MSVR acierta en los 4 primeros tests y en los 3 restantes falla. En cuanto a la estimación, RNRUP existe un caso en el que incluso estima con un 100% de acierto, el resto la estimación es bastante buena. El método MSVR es menos preciso a la hora de estimar el valor de la QoE. El hecho de que existan más aciertos en los cuatro primeros casos es porque en el corpus existen casos muy parecidos a éstos y el método de estimación los ha aprendido bien.

6.4.5 Conclusiones

Para concluir decir que hemos planteado un sistema de gestión para garantizar la QoE basado en métodos de inteligencia artificial para la estimación de la QoE. Dado unos parámetros de entrada basados en diferentes parámetros de red, el sistema detecta qué formato de codificación da un valor estimado de QoE más alto y lo selecciona para la transmisión.

En esta sección hemos llevado a cabo el estudio para el desarrollo del método de estimación. Hemos analizado cada uno de los parámetros de entrada: jitter, retardo, paquetes perdidos y ancho de banda para cada uno de los formatos de codificación y para diferentes condiciones de red durante la transmisión. Las pruebas realizadas nos han servido para demostrar que cada códec se comporta de una forma diferente dependiendo de las condiciones de red. A partir de estos datos hemos creado un corpus con el que hemos llevado a cabo la configuración de los métodos de estimación. Al finalizar se han incluido en un sistema inteligente de gestión con el que hemos realizado un test para detectar qué formato de codificación maximiza el valor de QoE. El que mejores resultados ha dado ha sido el método de estimación basado en redes neuronales.

Los resultados de este método se aproximan mucho a los valores reales de estimación de la QoE obtenidos mediante valoraciones subjetivas. Los métodos subjetivos de evaluación son muy costosos por lo que nuestro método de predicción podría ser una alternativa muy buena, de importante consideración.

En futuros trabajos se podría mejorar el sistema aplicando técnicas de Principal Component Analysis (PCA) por ejemplo, con el fin de eliminar aquellas características que no nos ofrecen información relevante. También se podrían probar otras características como el tiempo de jitter, etc.

6.5 Conclusiones

En este capítulo hemos descrito los principales procesos realizados por el sistema de gestión. El sistema diferencia tanto el entorno de red como el tipo de usuario en la entrega del servicio IPTV y actúa en consecuencia.

En la primera sección de este capítulo hemos estudiado la gestión en entornos de red cableados. Hemos comprobado cómo la QoS influye en la QoE analizando la calidad de vídeo recibida en diferentes condiciones de red. En base a ello hemos planteado un algoritmo de gestión basado en la monitorización de la red con el fin de estimar la QoE. En el caso de que los valores estimados no superen un umbral, se intenta solucionar en primer lugar realizando cambios en la red. Si no es posible de esta forma se lleva a cabo transcodificación. La transcodificación se puede realizar de dos formas: heurística o inteligente. En esta sección describimos la forma heurística.

En la siguiente sección hemos estudiado la gestión en entornos de red inalámbricos. Cuando el cliente está en un entorno inalámbrico, la prioridad del sistema de gestión es unir al cliente a la red que mejor QoE proporcione. Para ello el sistema de gestión realiza tres procesos: análisis de los parámetros de QoS, búsqueda de redes inalámbricas y cambio de red. En el caso de que no se solucione así, el sistema de gestión lleva a cabo los procesos normales de realización de cambios en la red y en su defecto transcodificación.

El siguiente proceso de gestión que hemos analizado en este capítulo es para clientes con dispositivos móviles. Cuando el acceso al servicio de IPTV se realiza desde un dispositivo móvil, el controlador de vídeo toma muy en cuenta las características del dispositivo para seleccionar el formato de codificación del flujo de vídeo que se va a transmitir. Características como: la resolución de pantalla y limitaciones de hardware, van a condicionar las características del flujo de vídeo. Para explicar cómo se lleva a cabo el proceso de gestión hemos desarrollado tanto un algoritmo de gestión cliente-servidor como un protocolo de comunicación cliente-servidor. Para llegar a este planteamiento primeramente analizamos en profundidad los parámetros de QoS en transmisiones de flujo de vídeo para dispositivos móviles en entornos de redes UMTS y WIFI.

En la última sección de este capítulo, hemos llevado a cabo el desarrollo y aplicación del método inductivo de estimación que integraremos en el sistema de gestión. El sistema de gestión utilizará el método de estimación para dos procesos: para estimar la QoE y para seleccionar el formato de codificación en el proceso de transcodificación inteligente. En base a los parámetros de QoS y al formato de codificación de la transmisión el sistema estima la QoE usando un método de regresión inductivo. Para la transcodificación, el sistema selecciona aquel formato de codificación que maximiza la QoE, dado unos parámetros de QoS conocidos. El método inductivo utilizado para ambos casos es el mismo. Para desarrollar el

método hemos creado un corpus de entrenamiento y de test a partir de vídeo clips 3D/ST. Para ello, hemos realizado transmisiones de flujo de vídeo con estos formatos en distintas condiciones de red. Hemos tomado medidas de los parámetros de QoS y hemos configurado cada uno de los vectores de características que conforman el corpus. Para etiquetar cada una de las muestras hemos llevado a cabo valoraciones subjetivas y objetivas de cada vídeo clip. El promedio de ambas medidas nos ha dado como resultado un valor estimado de la QoE y que hemos utilizado como etiqueta. Una vez definido el corpus, hemos analizado que método inductivo de estimación se ajustaba más al problema. En base a los resultados obtenidos, hemos concluido que RNRUP es el método que utilizaremos en el sistema de gestión.

Por último, cabe decir que en este capítulo hemos hecho referencia a algunos de nuestros estudios, publicados en diferentes actas de congresos y revistas internacionales, como son: (García y col. 2009), (Lloret y col. 2011b), (Lloret y col. 2011c), (Lloret y col. 2012), (Bri y col. 2013) y (Lloret y col. 2013). Con ello, el principal objetivo ha sido el desarrollo de un sistema de gestión para la transmisión de flujo IPTV que garantice la QoE del usuario final.

Capítulo 7

Software de gestión

En la sección 7.1 de este capítulo, hemos realizado una introducción centrándonos en el lenguaje de programación Java ya que ha sido el utilizado para la implementación del software. En la sección 7.2 hemos descrito la estructura del software basándonos en un diagrama de clases. En la sección 7.3 hemos descrito el uso del software de gestión. En la sección 7.4 hemos llevado a cabo la integración del software desarrollado en un entorno de laboratorio para comprobar su aplicabilidad.

7.1 Introducción

Para la implementación del sistema de gestión hemos utilizado el lenguaje de programación Java. La elección de este lenguaje de programación se debe a la enorme potencia que ofrece la Programación Orientada a Objetos. Entre sus características se encuentra que permite que un programa esté estructurado en módulos y por tanto, exista un cambio fácil en cualquiera de las partes.

Java es una plataforma virtual de software desarrollada por Sun Microsystems. Los programas creados en ella pueden ejecutarse sin cambios en diferentes plataformas. Java consta de las siguientes partes: el lenguaje de programación, la máquina virtual (Java Virtual Machine (JVM)) que permite que el programa se pueda ejecutar sobre cualquier plataforma y el Application Programming Interface (API) Java.

Los programas en Java generalmente son compilados a un lenguaje intermedio llamado bytecode, que luego son interpretados por una la JVM que interpreta el código convirtiéndolo a código particular de la CPU utilizada. Cualquier aplicación que se desarrolle, se apoya en un gran número de clases preexistentes que forman parte del propio lenguaje (el API o API de Java). Java es un lenguaje orientado

a objetos muy seguro y robusto que permite tener múltiples tareas ejecutándose simultáneamente.

El entorno de desarrollo utilizado para la programación y testeo del código fuente ha sido el entorno de desarrollo integrado (Integrated Development Environment (IDE)) ofrecido por Netbeans. Netbeans es una de las Herramientas más sólidas para desarrollo para diseño de aplicaciones en la plataforma Java 7. Está orientada, principalmente, al campo profesional, como consecuencia de su buena disposición en las áreas de tratamiento de datos y preparación para entornos distribuidos. Netbeans es un entorno de desarrollo multiplataforma ya que corre en diversos sistemas operativos, entre los cuales se encuentra Microsoft Windows, Unix y Linux.

7.2 Estructura del software

En este apartado describiremos la estructura del software. Para ello, en primer lugar mostraremos un diagrama en lenguaje unificado de modelado (Unified Modeling Language (UML)) en el que se especifica, visualiza y describe el esquema del software de gestión tal como podemos observar en la figura 7.1.

Como podemos observar en la figura 7.1 el programa tiene dos clases principales, que son: `FrameInicio` y `FrameGestion`. La clase `FrameInicio` se corresponde con la primera pantalla del programa en la que se testea el estado de la conexión y posibilidad de monitorización del dispositivo que se va a gestionar. Si el test es positivo se puede pasar a la siguiente pantalla correspondiente a la clase `FrameGestion` en la que se monitoriza y se gestiona el dispositivo en cuestión. A continuación explicaremos con más detalle la estructura de cada una de estas clases.

7.2.1 Pantalla de test

En la pantalla de test las principales clases que se utilizan son: `FrameInicio` y `Test`. La clase `Test` utiliza las clases `Ping` y `SnmpGestion` para enviar un ping y comprobar la conexión y para realizar una consulta SNMP y comprobar la posibilidad de monitorización del dispositivo.

FRAMEINICIO:

La clase `FrameInicio` hereda de la clase `JFrame` que hemos utilizado para implementar el interfaz de la pantalla de inicio o test. Tal como podemos ver en la figura 7.2, la pantalla de test se divide en cuatro partes principalmente definidas por cuatro `JPanel`: la parte superior correspondiente a la parte de registro, la parte inferior a ésta corresponde a la parte de conexión, la siguiente correspondiente a

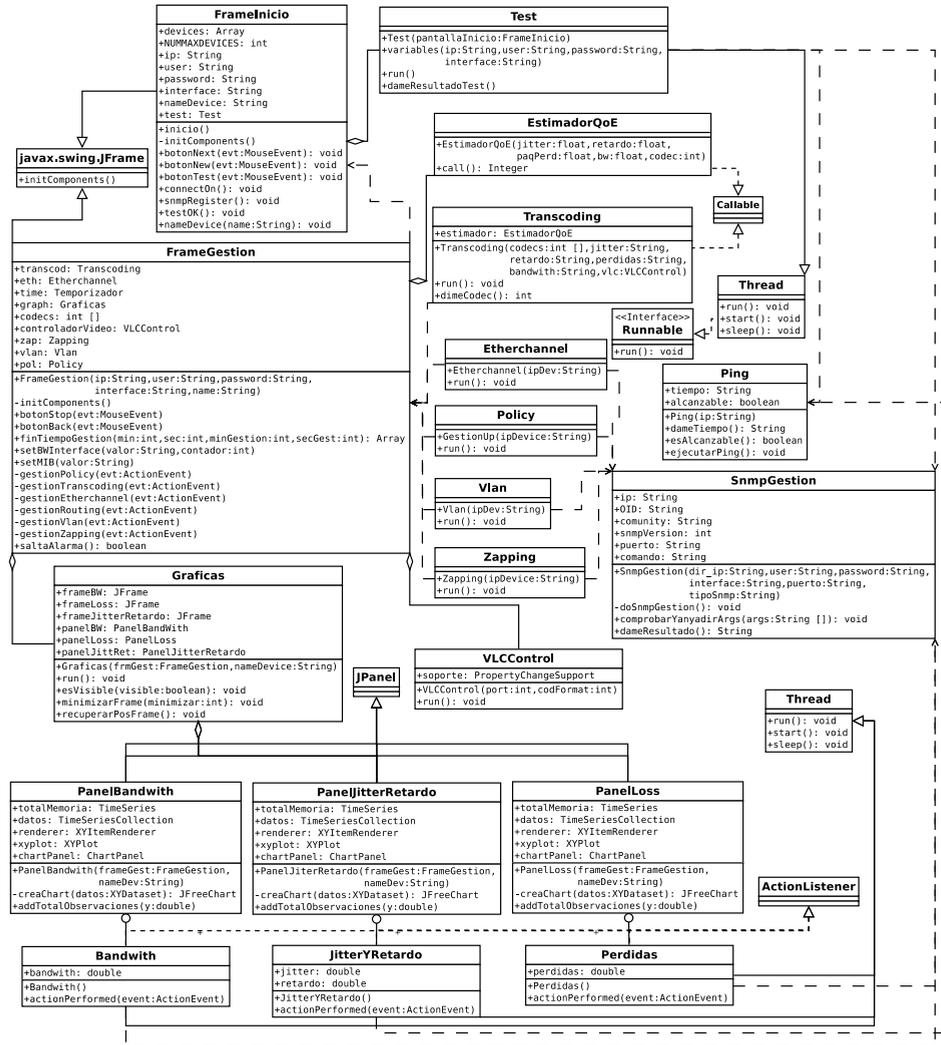


Figura 7.1: Diagrama de clases uml.

la de inicio y resultado del test y por último la parte inferior de la pantalla en la que se puede pasar de pantalla o iniciar una nueva configuración de gestión.

En la parte de registro existen dos campos de texto, etiquetados como “User” y “Password”, en los que se introduce el nombre de usuario y la contraseña que utilizaremos para las consultas SNMP. Esta información es necesaria ya que utilizamos SNMP -v3 y con esta versión es necesaria la seguridad de las consultas.

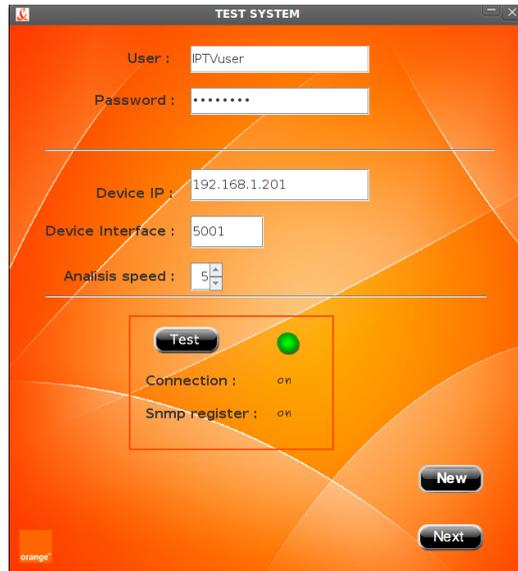


Figura 7.2: Pantalla de inicio del programa de gestión donde se introducen los datos de administración.

En la parte de conexión hemos integrado dos campos de textos y una caja combo. En el primer campo, “Device IP”, se debe introducir la dirección IP del dispositivo que se quiere gestionar. En el siguiente, “Device interface”, añadiremos el número del interfaz o puerto del dispositivo. Por último, en la caja combo de “Analysis speed”, introduciremos el intervalo en segundos en el que se realizará cada proceso de monitorización.

La siguiente parte de la pantalla es la de conexión. En esta parte tenemos un botón, junto a él un botón radial y debajo dos etiquetas una debajo de la otra. El botón “Test” sirve para iniciar el test de la conexión. Al lado hay una etiqueta con un icono en forma de led que nos indica si el test ha sido positivo o no, poniéndose en verde si lo ha sido y en rojo si no. Para determinar qué parte del test ha sido correcta disponemos de las dos etiquetas siguientes. La primera, etiquetada como “Connection”, nos indica si la conexión ha sido establecida. Si ha sido así la etiqueta mostrará el valor de on. Por el contrario, si no se ha podido establecer la conexión la etiqueta mostrará el valor de off. Lo mismo ocurre con la siguiente, etiquetada como “Snmp register”, en la que comprobaremos si es posible establecer una conexión snmp con el dispositivo a gestionar.

Finalmente, en la parte inferior de la pantalla hay dos botones “New” y “Next”. Con el botón “New” podemos iniciar una nueva configuración y test de conexión de un nuevo dispositivo. Con el botón “Next” pasaremos a la siguiente pantalla de monitorización de gestión del dispositivo configurado.

TEST:

La clase `FrameInicio` contiene una clase `Test`. La clase `Test` hereda de la clase `Thread` que implementa a la clase `Runnable`.

La clase `Test` se ejecuta al pulsar el botón “Test” de la pantalla de inicio. Al ejecutarse realiza un ping y una consulta SNMP a la IP del dispositivo. Para realizar el ping utilizamos la clase `Ping`. El comando utilizado para llevar a cabo el ping es el siguiente:

```
ping -c 1 -s 1492 ip
```

donde *ip* es la IP del dispositivo que queremos gestionar.

Para realizar la consulta SNMP utilizamos la clase `SnmpGestion`. La consulta que se realiza mediante la clase `SnmpGestion` sobre la Management Information Base (MIB) del dispositivo es del siguiente tipo:

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.1.5
```

donde *ip* es la IP del dispositivo que queremos gestionar, *user* el nombre de administrador y *password* la contraseña.

El resultado del test lo obtendremos llamando al método “`dameResultadoTest`” de `Test` desde la clase `FrameInicio`.

PING:

La clase `Ping` tiene dos funciones principalmente: analizar si existe conexión con el dispositivo que se pretende gestionar y obtener el tiempo que tarda en realizarse dicho ping con el fin de extraer el tiempo de jitter y retardo.

La clase `Ping` se compone de un constructor cuyo parámetro de entrada es la IP del dispositivo. Además dispone de tres métodos con las siguientes funciones. Con el método “`ejecutarPing`” se lleva a cabo el ping. Con el método “`esAlcanzable`” podemos saber si el dispositivo es alcanzable o no y por tanto si hay conexión o no. Por último, con el método “`dameTiempo`” podemos saber el tiempo medio que ha tardado en realizarse el ping.

SNMPGESTION:

Utilizamos la clase `SnmppGestion` para realizar tareas de monitorización sobre los dispositivos a gestionar mediante mensajes SNMP. Las tareas de gestión pueden ser de dos tipos: consultas o configuración. A través de las consultas obtenemos información de los dispositivos y mediante la configuración cambiamos aspectos de la configuración del dispositivo.

Los parámetros de entrada del método constructor de la clase `SnmppGestion` son: un `String` con la IP del dispositivo, un `String` con el password para tener acceso a la gestión del dispositivo, un `String` con el puerto o interfaz del dispositivo que queremos gestionar y por último un `String` indicando el tipo de mensaje SNMP (`snmpwalk` para realizar una consulta y `snmpset` para realizar un cambio en la configuración del dispositivo).

A parte del constructor, la clase `SnmppGestion` dispone de un método, denominado “doSnmppGestion”, desde el que se llevará a cabo la ejecución de la gestión SNMP. El método “comprobarYanadirArgs” configura los argumentos del mensaje SNMP. Por último, en el caso de que la gestión SNMP haya sido una consulta podremos obtener el resultado llamando al método “dameResultado”.

7.2.2 Pantalla de monitorización y gestión

En la pantalla de monitorización y gestión las principales clases que se utilizan son: `FrameGestion`, `Graficas`, `EstimadorQoE`, `Transcoding`, `VLCControl`, `Etherchannel`, `Policy`, `Vlan` y `Zapping`. De `Graficas` cuelgan las clases `PanelBandwith`, `PanelJitterRetardo` y `PanelLoss` las cuales tienen incluidas las clases `Bandwith`, `JitterYRetardo` y `Perdidas` respectivamente. Desde `Bandwith`, `JitterYRetardo` y `Perdidas` se realiza todo el proceso de monitorización de la red en cuanto a parámetros de red como: jitter, retardo, paquetes perdidos y ancho de banda.

FRAMEGESTION:

La clase `FrameGestion` hereda de la clase `JFrame` que hemos utilizado para implementar el interfaz de la pantalla de monitorización y gestión. Tal como podemos ver en la figura 7.3, en la pantalla de monitorización y gestión podemos diferenciar dos partes, una parte (derecha de la pantalla) donde están las gráficas en las que se muestran los resultados de monitorización por cada intervalo de tiempo de gestión y otra (izquierda de la pantalla) en la que se muestran los valores exactos de monitorización, los procesos de gestión elegidos y cuáles se están llevando a cabo. Cada una de las partes definidas está incluida en un `JPanel`.

La parte de las gráficas consta de tres `JPanel` uno por cada una de ellas, es decir, uno para la gráfica de jitter y retardo, otro para las pérdidas y otro para el ancho de banda. Las gráficas son de la clase `JFreeChart` pero cada una de un tipo distinto



Figura 7.3: Pantalla de gestión del software

con el fin de ayudar en la interpretación de los datos. Para el jitter y retardo los datos son representados según XYLineChart, es decir con líneas, para las pérdidas XYBarChart, o sea con barras y por último XYAreaChart para el ancho de banda, es decir con áreas.

Cada valor representado en la gráfica se muestra de forma numérica mediante etiquetas definidas por los siguientes nombres: “Bandwith interface” para el ancho de banda del canal disponible, “Jitter” para el jitter, “Delay” para el retardo, “Loss” para las pérdidas y “Bandwith used” para el ancho de banda del canal utilizado. En este mismo JPanel hay dos botones: “Stop” | “Start” y “Back”. Con “Stop” | “Start” detenemos o iniciamos el proceso de gestión respectivamente. Con “Back” regresáramos a la pantalla de test.

En la parte superior-izquierda del programa diferenciamos dos partes. A la derecha, tenemos una serie de JRadioButton desde dónde seleccionaremos el proceso de gestión que deseamos que se lleve a cabo. Al lado de cada JRadioButton hay una etiqueta que identifica el proceso de gestión. Por último, cada uno de estos procesos tiene un led que se activa en el caso de que se esté ejecutando el proceso de gestión. La etiqueta de cada uno de los procesos es: “In Queue Policy”, “Out Queue Policy”, “Transcoding”, “Etherchannel”, “Routing”, “Vlan” y “Decrease zapping time”. En el caso de que se hayan elegido todos los métodos de gestión el orden de ejecución es el siguiente: primero Zapping (si el tiempo de zapping es superior a 300ms), Policy, etherchannel y por último Vlan. Si tras realizar todos estos procesos continúa sin solucionarse el problema se llevará a cabo transcoding. Para llevar a cabo cada

uno de estos procesos de gestión se realiza la llamada a la correspondiente clase desde FrameGestion. Estas clases son: Transcoding, Etherchannel, Policy, Vlan y Zapping. A la izquierda, hay un área de texto en la que se va indicando cada uno de los procesos de gestión que se están ejecutando.

GRAFICAS:

La clase Graficas hereda de la clase Thread. Contiene tres clases que son: PanelBandwith, PanelJitterRetardo y PanelLoss. La clase FrameGestion contiene una clase Graficas para representar en gráficas los valores de jitter, retardo, pérdidas y ancho de banda utilizado.

PANELBANDWITH Y BANDWITH:

La clase PanelBandwith hereda de la clase JPanel. Esta clase principalmente se encarga de crear y diseñar una gráfica así como de renderizar los datos correspondientes al ancho de banda. Para la obtención de estos datos utilizamos la clase Bandwith la cual está incluida en la clase PanelBandwith. Para la creación y diseño de estas gráficas utilizamos tres métodos principalmente: el constructor desde el cual le pasamos como argumentos la pantalla en la que lo vamos a colocar (FrameGestion) y el nombre del dispositivo mediante un String. Con el método “crearChart” creamos y diseñamos la gráfica. Por último, con el método “addTotalObservaciones” ajustamos el total de memoria utilizada en la representación de los datos y de esta forma liberar memoria del programa. Este valor lo pasaremos mediante un argumento de tipo double con valor en milisegundos.

La clase Bandwith implementa la clase ActionListener. De esta forma en la implementación del método actionPerformed ejecutaremos un proceso utilizando la clase Process con el que obtendremos el valor del ancho de banda utilizado y cada vez que obtengamos este valor se representará en la gráfica. Para la obtención del ancho de banda llamaremos a la clase SnmpGestion con la que realizaremos diferentes consultas sobre la MIB del dispositivo que estamos gestionando y así obtener los valores necesarios para poder calcular la utilización del canal mediante la ecuación:

$$BW = ((ifInOctets(n) - ifInOctets(n - 1) + ifOutOctets(n) - ifOutOctets(n - 1)) * 8 * 100) / (intervalo * ifSpeed) / 2 \quad (7.1)$$

Donde, *intervalo* es el intervalo de tiempo entre cada consulta, *n* el número de consulta realizada, *ifSpeed* es una estimación del ancho de banda actual de la interfaz en bits por segundo, *ifInOctets* es el número total de octetos recibidos en la interfaz y *ifOutOctets* es el número total de octetos transmitidos por la interfaz.

Como podemos observar en la Ecuación 7.1 necesitamos conocer los valores de `ifOutOctets`, `ifInOctets` y `ifSpeed` en cada intervalo de tiempo. La consulta que se lleva a cabo para obtener esta información por cada intervalo es:

```
#ifInOctets
```

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.2.2.1.10.interface
```

```
#ifOutOctets
```

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.2.2.1.16.interface
```

```
#ifSpeed
```

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.2.2.1.5.interface
```

donde *ip* es la IP del dispositivo que queremos gestionar, *user* el nombre de administrador, *password* la contraseña e *interfaz* es el número de puerto del canal que queremos gestionar. Como podemos observar el nombre del objeto de la MIB lo hemos identificado por su OID correspondiente.

PANELJITTERRETARDO Y JITTERYRETARDO:

La clase `PanelJitterRetardo` tiene la misma estructura que la clase `PanelBandwith` con la diferencia de que los datos que representa se corresponden con los del jitter y retardo, que el aspecto de la representación de dichos datos es diferente y que la clase que utiliza para la obtención de estos datos es `JitterYRetardo`.

La clase `JitterYRetardo` implementa la clase `ActionListener`. En base a ello, en la implementación del método `actionPerformed` ejecutamos un proceso, utilizando la clase `Process`, para el cálculo de los valores de jitter y retardo. Cada vez que adquirimos este valor se representa en la gráfica. Para el cálculo del jitter y retardo llamamos a la clase `Ping` con la que realizamos un ping sobre el dispositivo que estamos gestionando y a partir del tiempo de respuesta obtenemos dichos parámetros de red. A partir del valor medio del tiempo de respuesta obtenemos el retardo. El jitter, es la diferencia entre dicho retardo y el retardo de la respuesta anterior a esta.

PANELLOSS Y PERDIDAS:

La clase `PanelLoss` tiene la misma estructura que la clase `PanelBandwith` con la diferencia de que los datos que representa se corresponden con los de las pérdidas de paquetes, que el aspecto de la representación de dichos datos es diferente y que la clase que utiliza para la obtención de estos datos es `Perdidas`.

La clase `Perdidas` implementa la clase `ActionListener`. En base a ello, en la implementación del método `actionPerformed` ejecutamos un proceso, utilizando la clase `Process`, para el cálculo de las pérdidas en el canal. Cada vez que adquirimos este valor se representa en la gráfica. Para el cálculo de las pérdidas llamaremos a la clase `SnmpGestion` con la que realizaremos diferentes consultas sobre la MIB del dispositivo que estamos gestionando. Con ello, conseguimos los valores necesarios para poder calcular las pérdidas mediante la ecuación:

$$Perdidas_T = \frac{perdidas(n) - perdidas(n - 1)}{ifSpeed * intervalo} \quad (7.2)$$

Siendo,

$$perdidas(n) = ifOutUcastPkts * ifMtu * 8 \quad (7.3)$$

Donde, *intervalo* es el intervalo de tiempo entre cada consulta, *n* el número de consulta realizada, *ifOutUcastPkts* son el número total de paquetes que los protocolos de nivel superior han solicitado transmitir, que no son del tipo multicast o broadcast, incluyendo los que han sido descartados o no enviados, *ifMtu* es el tamaño del paquete más grande que puede ser enviado o recibido en la interfaz y se especifica en octetos y *ifSpeed* es una estimación del ancho de banda actual de la interfaz en bits por segundo.

Como podemos observar en la Ecuación 7.2 necesitamos conocer los valores de `ifOutUcastPkts`, `ifSpeed` y `ifMtu` en cada intervalo de tiempo. La consulta que se lleva a cabo para obtener esta información por cada intervalo es:

#ifSpeed

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.2.2.1.5.interface
```

#ifOutUcastPkts

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.2.2.1.17.interface
```

#ifMtu

```
snmpwalk-v3 -u user -l authNoPriv -a MD5 -A password ip 1.3.6.1.2.1.2.2.1.4.interface
```

donde *ip* es la IP del dispositivo que queremos gestionar, *user* el nombre de administrador, *password* la contraseña e *interfaz* es el número de puerto del canal que queremos gestionar. Como podemos observar el nombre del objeto de la MIB lo hemos identificado por su OID correspondiente.

ETHERCHANNEL, POLICY, VLAN Y ZAPPING:

Las clases Etherchannel, Policy, Vlan y Zapping heredan de la clase Thread. En cada una de estas clases, el subproceso que se lleva a cabo al iniciarse desde la clase FrameGestion consiste básicamente en realizar un cambio en la configuración del router utilizando la clase SnmpGestion. El comando snmpset que se utiliza para ello es del siguiente tipo:

```
snmpset-v3 -u IPTVuser -l authNoPriv -a MD5 -A iptvpass ip .1.3.6.1.4.1.9.2.1.53.192.168.1.100
s "nuevaConfiguracion"
```

Al ejecutarse este comando, lo que va a hacer el router es cargar un fichero desde el servidor con la nueva configuración. Cada configuración está guardada en un fichero diferente. Por tanto, para cada proceso de gestión lo único que cambia sobre dicho comando es el nombre de "nuevaConfiguracion" por el del fichero correspondiente. Para identificar el router, en el constructor de la clase le pasaremos la IP del dispositivo. Los cambios que se van a realizar sobre el router con cada uno de los procesos de gestión son básicamente:

- Etherchannel:** Con Etherchannel se unen dos canales en uno, aumentando de esta forma el ancho de banda disponible.
- Policy:** Hay que diferenciar entre la Policy In y Policy Out. Con Policy In se realiza un cambio en las políticas del router con respecto a los paquetes de entrada al interfaz y con Policy Out con respecto a los de salida. El cambio de política consiste básicamente en dar una mayor prioridad a los paquetes rtp para IPTV sobre el resto.
- Vlan:** Con Vlan reagrupamos los grupos multicast con el fin de evitar congestión en la red.
- Zapping:** Con Zapping cambiamos el tiempo de jitter del canal con el fin de evitar que se de este problema.

ESTIMADORQoE:

La clase EstimadorQoE implementa la clase Callable. A diferencia de la clase Runnable, con la clase Callable podemos obtener un valor como resultado del proceso. Este resultado determinará si se activa una alarma y como consecuencia de ello el inicio de un proceso de gestión con el fin de mejorar la estimación de la QoE.

El proceso de estimación que se lleva a cabo en la clase EstimadorQoE es el siguiente: al pulsar el botón "start" desde la clase FrameGestion se inicia un bucle cuya finalización dependerá del valor obtenido con EstimadorQoE. Con cada iteración construimos una nueva clase EstimadorQoE a la que le pasaremos los valores ac-

tuales de jitter, retardo, pérdidas, ancho de banda y códec. El resultado obtenido es un Integer, tal como podemos observar en el diseño, correspondiente al valor de estimación de la QoE. Para obtener este valor, la clase EstimadorQoE contiene una clase Process con la que ejecutaremos un script .bat. que ejecutaremos mediante la herramienta BATCHMAN de *SNNS* (Stuttgart Neural Network Simulator). Con este script cargaremos una red neuronal entrenada, le pasaremos la muestra que queremos entrenar y volcaremos el resultado en un fichero de texto. El resultado lo leeremos de este fichero de texto, lo transformamos en Integer y se lo enviamos a la clase FrameGestion. El comando que hemos utilizado para ejecutar el script desde java es el siguiente:

```
String com = "./batchman -q -f programa.bat"; # String de la llamada al programa .bat
Process proc = Runtime.getRuntime().exec(com,null,path);
```

El código que contiene el .bat es el siguiente:

```
red:= "RedNeuronal.net"
patronTest:= "muestraTest.pat"
loadNet(red) # cargamos la red neuronal entrenada
loadPattern(patronTest) # cargamos la muestra de test con los parámetros de QoS y el códec
setPattern(patronTest)
resfile:= "res"
saveResult(resfile, 1, PAT, FALSE, FALSE, "create") # guardamos el resultado de test
```

TRANSCODING:

La clase Transcoding implementa la clase Callable y contiene una clase EstimadorQoE con la que obtendremos la estimación de la QoE de cada uno de los códecs que se necesita testear. El proceso de transcoding se activa cuando no ha sido posible solucionar una degradación de la QoE con ninguno de los otros procesos de gestión de red activados.

Al activarse el proceso de transcoding en primer lugar se contruye una clase Transcoding a la que le pasaremos los siguientes parámetros: jitter, retardo, pérdidas, ancho de banda, un array con los códecs a testear y la clase VLCCControl con la que se está realizando la transmisión. A continuación se obtiene por cada códec la estimación de la QoE utilizando la clase EstimadorQoE. Si el códec con mejor estimación de QoE obtenida mejora la del códec actual se iniciará el proceso de transcodificación. Para ello, se realiza una llamada al método de "transcoding" de la clase VLCCControl y se cambia de codificación del vídeo que se está transmitiendo.

VLCCCONTROL:

La clase VLCCControl hereda de la clase Thread. Con esta clase ejecutamos un script con el que iniciamos la transmisión del vídeo utilizando el programa VLC

y también llevamos a cabo el proceso de transcodificación del vídeo durante la transmisión. El comando utilizado para la transmisión del vídeo es del tipo:

```
vlc -I rc -lua-config rc=host='localhost:5000'vvv -start-time=0 /Path/codificacionFormat  
-sout '#rtpdst=232.32.32.32,port=1234,mux=ts,t1=12'
```

siendo, *codificacionFormat* una variable cuyo valor se corresponde con el formato de codificación. Con esta variable especificaremos el factor de cuantificación y el códec de compresión de vídeo.

En el caso de que se vaya a realizar transcodificación le pasaremos a esta clase el nuevo formato de codificación extraído con la clase Transcoding. Seguidamente se ejecuta el script para transcodificación con el que se consulta primero el tiempo transcurrido en la transmisión de vídeo y a continuación se reinicia la transmisión de ese mismo vídeo en ese mismo instante pero con la nueva codificación. Para ello, se utiliza el mismo comando que acabamos de especificar pero indicando en *start-time* el tiempo de reinicio y en *codificacionFormat* el formato de codificación.

7.3 Descripción del funcionamiento del software

El software consta de dos partes fundamentalmente: una primera pantalla (figura 7.2) en la que el administrador introduce los datos para establecer la conexión entre la herramienta y los dispositivos de gestión y otra pantalla (figura 7.3) en la que se lleva a cabo los principales procesos de gestión como son: estimación de la QoE, monitorización, cambio de configuración de la red y transcodificación. El algoritmo de gestión en el que se basa el funcionamiento del programa se muestra en la figura 7.4.

En el algoritmo de la figura 7.4 vemos que hay una primera fase de petición de servicio por parte del usuario. Esta fase se realiza una vez se inicia la transmisión de vídeo, al conectarse el usuario al puerto desde el que se está llevando a cabo dicha transmisión. Para llevar a cabo la transmisión se utiliza el programa VLC desde el propio programa de gestión. La siguiente fase consiste en la clasificación del tipo de servicio. Para ello se necesita conocer el tipo de dispositivo, la codificación de la transmisión y el tipo de conexión. Toda esta información se extrae al finalizar el proceso de comprobación de conexión entre el dispositivo y el servidor llevado a cabo en la primera pantalla (figura 7.2). Al finalizar este proceso podemos pasar a la pantalla siguiente donde podremos iniciar la fase de gestión y monitorización de los dispositivos conectados. En base a los parámetros de QoS obtenidos con la monitorización de los dispositivos se realiza una estimación de la QoE. Si existe una degradación de la QoE inferior al umbral definido, se inicia el proceso de gestión con el fin de subsanar dicha degradación. Tras explicar cómo funciona el programa, describiremos como utilizarlo.

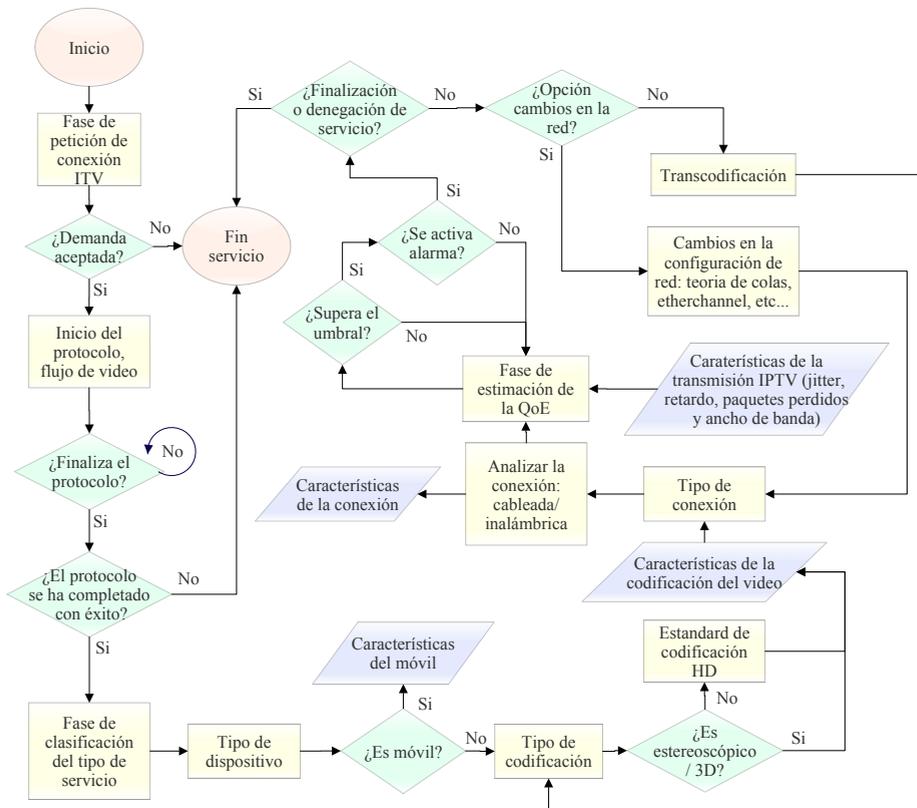


Figura 7.4: Algoritmo de gestión utilizado por el software

En primer lugar, en la primera pantalla se introducen los datos necesarios para establecer una conexión SNMP (Simple Network Management Protocol). Para ello deberemos rellenar las casillas, según la figura 7.5, con los siguientes datos:

1. Usuario
2. Contraseña
3. IP del dispositivo a gestionar
4. Puerto del dispositivo a gestionar

5. Intervalo del tiempo de gestión

The screenshot shows a window titled "TEST SYSTEM" with a dark orange background. It contains five input fields, each with a corresponding number on the right side:

- 1. User : IPTVuser
- 2. Password :
- 3. Device IP : 192.168.1.201
- 4. Device Interface : 5001
- 5. Analysis speed : 5

Figura 7.5: Datos necesarios para la conexión con el dispositivo a gestionar.

Para comprobar que los datos introducidos en la pantalla de test (figura 7.5) son correctos y establecer una conexión segura con los dispositivos monitorizados, se realiza un ping y una consulta SNMP del tipo *snmpwalk*. Mediante el ping se comprueba si la conexión es correcta y mediante la consulta SNMP se corrobora si es posible realizar consultas a la MIB de los dispositivos monitorizados. Para realizar el test los pasos que debemos seguir se muestran en la figura 7.6 y son los siguientes:

1. Pulsamos el botón “Test”.
2. Comprobamos que el resultado del test ha sido positivo. Si esto es así el color del led será verde y querrá decir que tanto el Packet Internet Groper (PING) como la consulta SNMP han funcionado correctamente. En caso de que haya fallado tanto el PING como la consulta SNMP o una de las dos el color del led será rojo. Para comprobar qué ha fallado comprobaremos tres y cuatro.
3. En el caso de que el test haya fallado comprobaremos primero el resultado del PING. Si el PING ha sido correcto aparecerá una etiqueta con el valor “on”. En el caso contrario el valor de la etiqueta será “off”.
4. Por último, comprobaremos el resultado de la consulta SNMP. Si ha sido correcta aparecerá una etiqueta con el valor “on”. En el caso contrario el valor de la etiqueta será “off”.

Si el resultado del test es correcto, se habrá establecido la conexión con el dispositivo, lo que permitirá al programa obtener características como: el tipo de

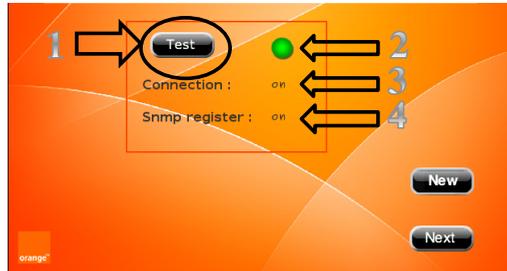


Figura 7.6: Proceso de comprobación de conexión con el dispositivo a gestionar.

dispositivo, el tipo de codificación y el tipo de conexión, siguiendo de esta forma el proceso descrito en el algoritmo de la figura 7.4. Esto es posible ya que el programa a través de los dispositivos monitorizados obtiene información de la red, de sus interfaces y de los propios dispositivos. En base a esta información se pueden determinar las características anteriormente nombradas. Cabe decir que por cada dispositivo que queramos monitorizar deberemos repetir todo el proceso anterior. Para ello deberemos pulsar el botón “New” (figura 7.6) y se nos abrirá una nueva pantalla de test que tendremos que completar con los datos del nuevo dispositivo.

Una vez establecida la conexión ya podremos iniciar el proceso de monitorización del dispositivo correspondiente. Para ello deberemos pasar a la siguiente pantalla (figura 7.3) pulsando el botón “Next” (figura 7.6). Al pulsar dicho botón se nos abrirá la pantalla de gestión en la que podemos diferenciar cuatro zonas tal como podemos apreciar en la figura 7.7:

1. Gráficas de monitorización de parámetros de red: jitter, retardo, pérdidas y ancho de banda.
2. Valores de monitorización de parámetros de red: jitter, retardo, pérdidas y ancho de banda.
3. Procesos de gestión. Selección y activación.
4. Descripción de los procesos de gestión que se están ejecutando.

Al pulsar el botón “start” (figura 7.7, zona 2) se inicia el proceso de estimación de la QoE, siguiendo el algoritmo de la figura 7.4. Para poder llevar a cabo dicha estimación antes es necesario monitorizar la red con el fin de obtener ciertos pa-

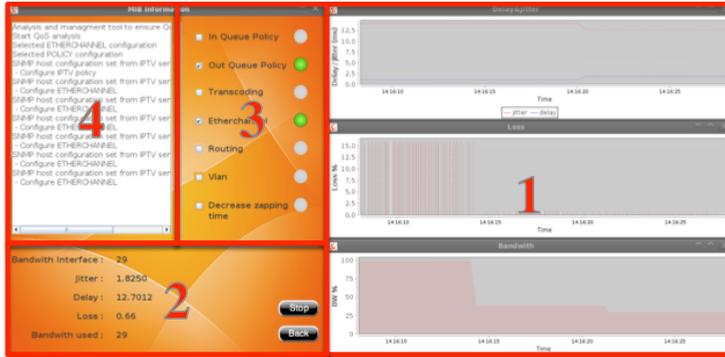


Figura 7.7: Proceso de comprobación de conexión con el dispositivo a gestionar.

rámetros de QoS como son: el jitter, el retardo, las pérdidas y el ancho de banda. Estos parámetros son representados gráficamente en la figura 7.7, zona 1, y numéricamente en la figura 7.7, zona 2. Para obtener dichos parámetros el programa realiza diferentes cálculos basados principalmente en información de la MIB del dispositivo y el tiempo de respuesta de un PING. Para el cálculo del jitter y el retardo el programa realiza un ping y a partir del tiempo de respuesta se extraen dichos parámetros. Para el cálculo de las pérdidas de paquetes y el ancho de banda consumido el proceso es un poco más complicado. Primero se realizan una serie de consultas a la MIB de los dispositivos monitorizados y a partir de esta información se calculan las pérdidas y el ancho de banda consumido. El nombre de los objetos pertenecientes a la MIB que se consultan son los siguientes:

- Interfaces.ifTable.ifEntry.ifInDiscards.ifIndex
- Interfaces.ifTable.ifEntry.ifInErrors.ifIndex
- Interfaces.ifTable.ifEntry.ifOutUnknownProtos.inIndex
- Interfaces.ifTable.ifEntry.ifInUcastPkts.ifIndex
- Interfaces.ifTable.ifEntry.ifInNUcastPkts.ifIndex

Para la estimación de la QoE el programa ejecuta un script .bat utilizando la herramienta BATCHMAN de *SNNS* (Stuttgart Neural Network Simulator). Este programa carga una red neuronal ya entrenada y a partir de los valores: jitter, retardo, paquetes perdidos, ancho de banda y formato de codificación con el que se está realizando la transmisión, se estima la QoE. Si el valor de la estimación de la QoE no supera un cierto umbral (que en el programa hemos fijado a 4) se inicia la activación de alarma (figura 7.4). Para que se active, debe darse este

caso al menos 3 veces en menos de 30 segundos. Si finalmente se activa la alarma, se iniciarán cambios en la configuración de la red según los procesos de gestión seleccionados y del orden de ejecución establecido en el programa. El orden de ejecución es el siguiente: Zapping (se ejecuta si el tiempo de zapping es superior a 300ms), Policy, etherchannel y por último Vlan. Si de esta forma continua habiendo una degradación en la QoE se realizará transcodificación. Para ello, el programa va probando cada uno de los códecs (compatibles con el tipo de transmisión y usuario) obteniendo una estimación de la QoE por cada uno de ellos mediante el proceso que acabamos de explicar. Aquel códec con el que se obtenga un valor estimado de la QoE mayor y que además supere el valor actual de la transmisión, será seleccionado para el proceso de transcodificación.

En la figura 7.7, zona 3, podemos seleccionar aquellos procesos de gestión que queramos que se ejecuten en el caso de que se produzca una degradación de la QoE. Conforme se van ejecutando los procesos de gestión se van mostrando en la figura 7.7, zona 4. Estos procesos son:

Política de colas entrada:	Aplica teoría de colas a los paquetes de entrada.
Política de colas salida:	Aplica teoría de colas a los paquetes de salida.
Transcodificación:	Realiza procesos de transcodificación.
Etherchannel:	Une más de un canal de transmisión para aumentar el ancho de banda.
Vlan:	Cambia configuraciones de las Vlan.
Reducir tiempo de zapping:	Reduce el tiempo de retardo de un enlace en concreto.

Tanto el proceso de estimación de la QoE como el de monitorización del dispositivo se inicia al pulsar el botón “start” de la segunda pantalla. Si deseáramos detener ambos procesos pulsaríamos de nuevo este botón. Para volver a la pantalla de test o inicio pulsaríamos el botón “back” que se encuentra en la figura 7.7, zona 2.

7.4 Integración del software de gestión

En esta sección mostraremos la integración de la herramienta de gestión en un entorno de laboratorio con el fin de demostrar su aplicabilidad. Para ello, en primer lugar hemos diseñado una topología de red que simule una arquitectura de red para una transmisión de IPTV tal como podemos ver en la figura 7.8.

Según la topología de la figura 7.8, el software de gestión iría instalado en el servidor controlador de vídeo correspondiente en este caso al *Servidor MAC*. Dicho servidor monitoriza los dispositivos que forman parte principalmente de la red de distribución y la red de acceso de una arquitectura de red IPTV. En este caso, estas secciones de red vienen representadas por los dos switches *SW1* y *SW2* y los dos

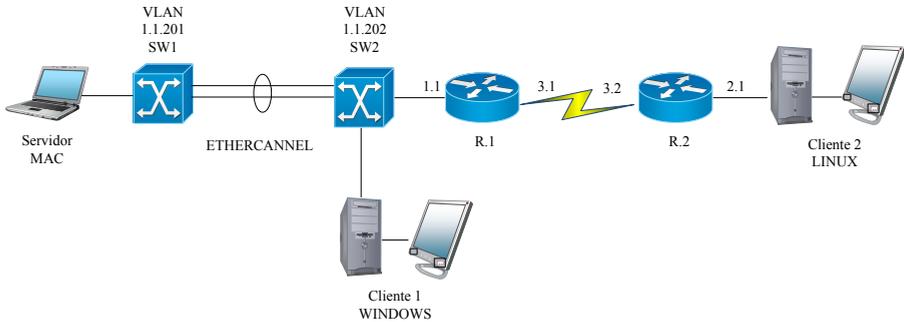


Figura 7.8: Topología de red tipo del banco de pruebas.

routers *R1* y *R2*. Cada uno de los canales entre el *SW1* y *SW2* es de 3MB. Para finalizar, hemos añadido 2 usuarios finales, uno cerca de la red de distribución con un sistema operativo Windows (*Cliente 1*) y otro en a la red de acceso con Linux (*Cliente 2*). El ancho de banda disponible por el cliente *Cliente 2* viene limitado por el enlace serial de los dos routers y es de 8MB, siendo un valor típico que podría ofrecer cualquiera de los proveedores de servicio en la actualidad a sus clientes. Cabe decir que en esta configuración hemos utilizado el servidor controlador de vídeo también como servidor de vídeo.

Al iniciar el programa, en primer lugar comprobará la conectividad con cada uno de los dispositivos de red que necesite monitorizar. Para ello introducimos el usuario y contraseña con la que se realizará una conexión segura para llevar a cabo las consultas SNMP con los dispositivos monitorizados. También introduciremos la IP del dispositivo a monitorizar y el puerto que utilizaremos para la conexión. Por último, introduciremos el intervalo de tiempo con el que se realizará cada consulta. Una vez introducidos los datos, pulsaremos el botón *Test* y si las respuestas del ping y SNMP son correctas tendremos la opción de pasar a la siguiente pantalla pulsando el botón *Next*. También tenemos la opción de realizar una nueva monitorización pulsando el botón *New* y repitiendo la configuración que acabamos de explicar.

Una vez establecida la conexión con los distintos dispositivos el controlador de vídeo inicia la monitorización de cada uno de ellos. Para ello, el programa inicia el cálculo de parámetros de red como son: el jitter, el retardo, las pérdidas y el ancho de banda consumido, tal como se ha explicado en la sección anterior. Para que el programa realice un determinado proceso de gestión debemos seleccionar

qué procesos deseamos se ejecuten por cada dispositivo. Entre los procesos se encuentran diferentes cambios de red y transcodificación.

Para este caso de estudio hemos seleccionado todos los procesos de gestión y un intervalo de tiempo de gestión de 5 segundos. Para ver cómo funciona la herramienta, desde el servidor de vídeo iniciamos una transmisión de un flujo de vídeo estéreo con un tamaño de 6MB. Además inyectamos otro tipo de tráfico a la red para darle más realidad a la simulación. Iniciamos el proceso de monitorización y estimación de la QoE. Los resultados obtenidos se muestran en las siguientes figuras.

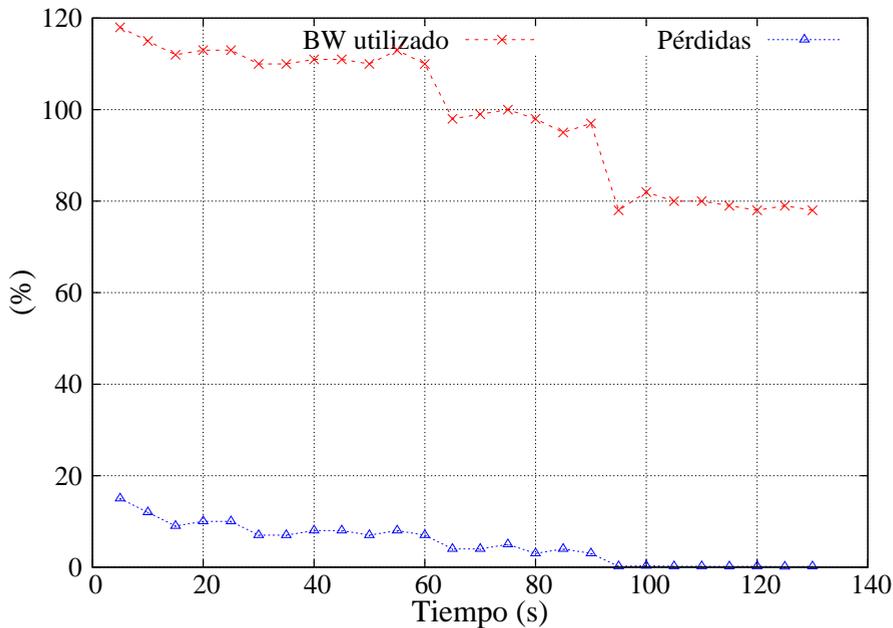


Figura 7.9: Ancho de banda utilizado y pérdidas de paquetes para el canal del cliente 2.

Cómo vemos en la figura 7.9 al inicio de la transmisión el ancho de banda utilizado supera el 100%. Esto provoca que haya un alto porcentaje de pérdidas tal como se ve en la misma figura. Debido a estas pérdidas de paquetes, la estimación de la QoE calculada por el programa es muy baja como podemos comprobar en la figura 7.10. Como esta situación se repite 3 veces en menos de 30s, da lugar a

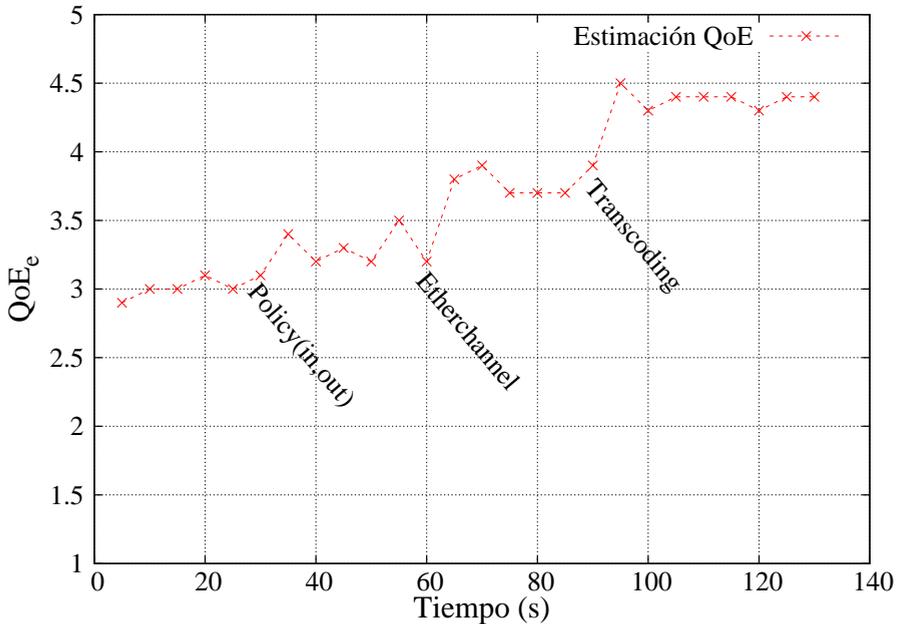


Figura 7.10: Estimación de la QoE para el canal del cliente 2.

que se active la alarma. Como hemos seleccionado todos los procesos de gestión el programa lo intenta solucionar en primer lugar realizando cambios en la red. Lo primero que comprueba es si el tiempo de jitter es superior a 300ms. Como no es así, pasa al siguiente proceso de gestión. Entonces intenta realizar cambios en las políticas del router para darle prioridad al tráfico RTP de IPTV. Como podemos observar en la figura 7.10, esta solución no es suficiente lo que provoca que continúe existiendo una degradación de la QoE. Como consecuencia, se activa de nuevo la alarma y el programa inicia un nuevo proceso de gestión. Como existe un único canal multicast, no es posible realizar grupos multicast por lo que pasa al siguiente proceso de gestión que es etherchannel. Existen dos canales entre el *SW1* y *SW2* de 3MB cada uno. Luego, es posible ejecutar dicho proceso de gestión entre ambos switches. El programa lo lleva a cabo convirtiendo de esta forma un canal de 3MB en uno de 6MB que es el ancho de banda que ocupa el vídeo. Con este último proceso de gestión la transmisión de vídeo ocupa prácticamente el 100% del ancho de banda disponible lo que continúa provocando algunas pérdidas tal como podemos observar en la figura 7.9 entre los segundos 60 y 90. Este último

cambio de red realizado tampoco es suficiente como podemos ver en la figura 7.10 entre los segundos 60 y 90. Por tanto, como última solución el programa lleva a cabo transcodificación. Para ello, el programa calcula qué formato de codificación 3D/ST maximiza el valor estimado de la QoE y que mejore el valor actual. Tras encontrarlo, el servidor cambia de formato de codificación de la transmisión lo que viendo los resultados obtenidos soluciona el problema. Con la nueva codificación, el ancho de banda del vídeo disminuye lo que hace que el ancho de banda utilizado ya no sea del 100 % y haya disminuido a valores del 80 %. A partir de estos momentos la QoE estimada supera el valor de 4, como vemos en la figura 7.10.

Podría ocurrir que las pérdidas fueran muy bajas pero que dado el entorno y el tipo de codificación (3D/ST) afectaran a la QoE. Con el método inductivo de estimación desarrollado podemos ser capaces de detectar estas situaciones. Sin embargo, con el método heurístico anteriormente desarrollado al haber unas pérdidas bajas y no superar el umbral no lo detectaría.

7.5 Conclusiones

En esta sección acabamos de presentar un software de gestión basado en el método inductivo de estimación. Hemos explicado su funcionamiento y hemos demostrado su aplicabilidad con un banco de pruebas.

El software de gestión consta de dos partes principalmente: una parte que podríamos definir de conexión y la otra de monitorización. En la parte de conexión en primer lugar comprobamos si es posible realizar la conexión y si lo es extraemos características como: tipo de dispositivo, tipo de conexión y tipo de codificación. En la parte de monitorización estimamos la QoE continuamente. En el caso de que supere un cierto umbral se inician los procesos de gestión basados en: cambios en la configuración de la red y si este no funciona, transcodificación.

Por tanto, podemos concluir que se ha cumplido uno de los objetivos de la tesis que era demostrar la viabilidad de un sistema de gestión automático que se ajuste a los cambios de la red. Una de las características a destacar del sistema es que es capaz de detectar cuando una degradación de la QoS en la red va a suponer una degradación de la QoE.

Capítulo 8

Conclusion

This chapter includes the conclusion that we have achieved with the work performed. Each previous chapter has included its own conclusion taking into account the work included in that chapter, but in this chapter we include the global conclusion. In section 8.1, we explain the objectives achieved in this PhD Thesis. In section 8.2 we describe the conclusions drawn from the research work and their implications. Section 8.3 shows the personal contributions of this work. In section 8.4 we list the publications derived from the research work. Finally, in section 8.5, we discuss future research lines.

8.1 Achievement of objectives

It has been possible to implement a management system based on artificial intelligence techniques for estimating the QoE. In this way, we succeeded the main objective of the project. In order to achieve our goal, we have carried out the following secondary objectives: build a corpus, set an inductive reasoning method to estimate the QoE and, finally, the integration of the estimation method in the management system. Achieving these goals has led us to conduct a thorough research in each one of the conducted studies, which have been published in international journals or conferences in order to corroborate our work.

8.2 Conclusion of the PhD Thesis

With the aim of building a corpus, we analyze the main QoS parameters that affect to the degradation of QoE. They are jitter, delay, packet loss and bandwidth. We perform standard, HD and 3D / stereoscopic video streams over wired (Ethernet), wireless (WIFI and WIMAX), and mobile networks. We use subjective assessment methods in order to evaluate the QoE of each transmission. With this information, we set the feature vectors and their corresponding labels thereby obtaining the samples to build the corpus.

The following operational objective was to find the inductive estimation method that best suits the problem of regression. We analyzed the models based on neural networks and support vector machines. Following the results, we decided that the product-unit neural networks are best suited. Then, we propose a QoE estimation management system which is based on both artificial intelligence techniques. The system has two main activation processes: alarms and service requests. An alarm occurs when it is estimated that the QoE is degraded. The video controller and routers are in charge of activating them. On the one hand, when routers detect that a jitter, delay or lost packets exceeds a threshold, then they will send a message to the video controller. The video controller is continuously monitoring the network and estimating the QoE using inductive estimation method. If the QoE value exceeds a threshold, an alarm will be also activated. On the other hand, the management system could also be activated if there is a service request. In this case, the controller analyzes the available resources, the type of device and the network environment; and, if accepted, the controller adjusts the delivery in order to guarantee the QoE at the customer side.

If an alarm is activated, the management system has two ways to solve the problem. First it tries to solve the problem by making network changes and if it does not work it starts to transcode the video. The transcode process can be done by heuristically or intelligently. When it is performed heuristically, the system takes into account parameters such as bandwidth or hardware limitations to select the appropriate encoding format. When it is performed intelligently, the system uses the inductive estimation method to select the appropriate encoding format to maximize the QoE value. The design of the management system fulfills the third objective. In conclusion, we have been able to develop a management system based on artificial intelligence techniques to guarantee the QoE at the end user side.

Note that one of the motivations of this PhD Thesis was to develop an objective method for estimating the QoE, with results similar to subjective QoE estimation methods, while avoiding its drawbacks. We can conclude that with the development of the inductive system we have started a new research line in this topic.

We have included a practical case that demonstrates the operation of the tool and its benefits. This part of the research project has been done into an agreement

with Orange, Spain, which was concluded few years ago. The development of this project has significant economic impact. In addition, it will provide important benefits at enterprise and social levels. On the one hand, the system will help the company to improve the management system IPTV service.

8.3 Personal contributions

This research PhD Thesis has helped me to deepen the study and application of different artificial intelligence techniques which allowed me to expand and anchor the knowledge acquired in this area. The fact of comparing and validating the proposed inductive estimation models to solve the real problem has helped me to increase my experience solving new problems. Moreover, I had to learn how to apply problem-solving methods to find a solution to each mishap occurred during the development of my proposal.

In addition, because this PhD Thesis has included several areas of knowledge, including telecommunications and artificial intelligence, this multidisciplinary research work has helped me to learn from all of them in order to achieve my goal.

The culmination of this PhD Thesis has provided me a great experience in both researching and developing projects at all phases. The project has been arduous yet rewarding given the obtained results. The development of this research project has also enhanced my ability to abstract general concepts from each part and perform properly each of them to reach the final goal. Finally, it has given me other important skills like teamwork and collaboration in my research group and with other research groups.

8.4 Publications derived from the research work

The work included in this dissertation has been published in several international conference proceedings and journals. They have been the following ones.

In (Canovas y col. 2010b) and (Atenas y col. 2010) we addressed the quantification and transcoding part. We conducted a thorough analysis of the influence of both processes in the IPTV QoE.

Another important issue in IPTV delivery is the analysis of the video streaming service and its influence on the QoE. In (Lloret y col. 2013) and (Canovas y col. 2009), we analyze this delivery process in a wireless environment and in (Canovas y col. 2012) we study how to improve it. In (Lloret y col. 2011c) and (Canovas y col. 2010a), we analyze the 3D/Stereoscopic video streaming in a wired environment. In (Lopez-Herreros y col. 2015), we propose a video delivery system for heterogeneous

IP networks using HTML5. In (Jimenez y col. 2015), we present a new algorithm to improve the QoE.

We also carried out the study and development of a management system to deliver an IPTV service in any type of network environment and video coding to guarantee enough QoE. The publications directly related to the development of this management system are the following ones. In (Lloret y col. 2012), we propose a management system for mobile environments. In (Lloret y col. 2011b), we analyze it in wired environments. In (Lloret y col. 2011a) we analyze the management system when delivering 3D/stereoscopic video flow in a wired environment and, finally, in (García y col. 2009), we analyze the management system taking into account the ubiquity of user devices.

One of the main aims of the management system is to know the characteristics of the end-user device. In (Bri y col. 2013) we explain how to obtain those characteristics using artificial intelligence techniques.

We would also like to mention that we have completed an article also directly related to this work under the title “A Cognitive Network Management System to Improve QoE in Stereoscopic IPTV”, which is still in the review process. This article presents a management system based on inductive prediction methods.

8.5 Future research lines

As a future work we propose to vary the type of features that build the corpus and use other techniques, such as PCA, to eliminate those features that are not necessary in order to achieve better results. In order to do this we analyze how other encoding and transmission parameters, such as resolution, frames per second, the zapping time, time transcoding, etc., affect the QoE and how we can apply them to the system. We also propose to increase the number of cases of IP video delivery and the number of encoding formats in order to expand the corpus and ensure the accuracy of the system for any type of video delivery. Regarding the encoding formats, we would like to extend the number of codecs in the system by adding privated and paid codecs as well as new 3D/stereoscopic codecs. Related to the video transmission, we would also like to increase the number of cases by adding new technologies such as LTE.

In our management system, in order to gather information from the network parameters such as: delay, jitter, packet loss and BW, we use SNMP agents. There are systems which allow obtaining this type of information from the headers of the RTP and RTCP packets. As a future research line, we propose a comparative study of both options.

As another future work, we want to apply the research obtained in this PhD Thesis to HTTP Adaptive Streaming, e.g. MPEG-DASH. Moreover, in order to further validate the benefits of our contributions we will increase the number of users in the performed test.

As a new research line, which has been started as a result of this PhD Thesis, we would like to develop a new method for estimating the QoE using artificial intelligence techniques and based on the values obtained from subjective methods for QoE estimation. This would allow us to reduce the costs involved the QoE is estimated using subjective methods.

Currently, in a wireless environment, the management system selects the wireless network based on the RSSI and QoE estimation. In a future work, we want to investigate how to apply artificial intelligence methods in order to make a more accurate selection, and investigate other parameters to determine which network is most suitable.

Finally, in relation to the developed tool, we plan to adapt the tool to a real environment in order to deploy the system any type of company. In order to adapt this tool we should analyze new parameters such as how fast are the estimation methods, because we didn't take care of it in the laboratory, but they could be critical in a real environment. We will also adapt the developed tool to learn from new environmental conditions. That is, it will be able to learn automatically from new transmission and coding cases.

Glosario

- ACR** Absolute Category Rating
- ACNS** Cisco Application and Content Networking System
- AQoS** Application-level Quality of Service
- AP** Access Point
- API** Application Programming Interface
- ASF** Advanced System Files
- AVI** Audio Video Interleave
- BW** Bandwidth
- CBR** Constant Bitrate
- CD** Compact Disc
- CMPQM** Color Moving Picture Quality Metric
- DCT** Discrete Cosine Transform
- DiffServ** Differentiated Service
- DS** Double Stimulus
- DSCQS** Double Stimulus Continuous Quality Scale
- DSIS** Double Stimulus Impairment Scale
- DSP** Digital Signal Processor
- DTV** Digital Television
- DVB** Digital Video Broadcasting

DVB-H Digital Video Broadcasting Handheld
DVD Digital Versatile Disc
EvalVid Video Quality Evaluation Tool
fps frames per second
FR Full Reference
GB Gigabyte
GHz Gigahertz
GOP Group of Pictures
HD High Definition
HDTV High Definition Television
HSDPA Speed Downlink Packet Access High
HTTP Hypertext Transfer Protocol
H.264/AVC H.264/Advanced Video Coding
IA Inteligencia Artificial
IDE Integrated Development Environment
IFC Information Fidelity Criterion
IGMP Internet Group Management Protocol
IMS IP Multimedia Subsystem
IntServ Integrated Service
IP Internet Protocol
IPTV Televisión sobre IP
ISP Internet Service Provider
ITU International Telecommunication Union
JMVC Joint Multiview Video Coding
JPEG Joint Photographic Experts Group
JVM Java Virtual Machine

JVT Joint Video Team

Kbps Kilobits per second

KHz Kilohertz

LAN Local Area Network

LCD Liquid Crystal Display

LTE Long Term Evolution

MBMS Multimedia Broadcast Multicast Service

Mbps Megabits per second

MDI Media Delivery Index

MIB Management Information Base

MIH Media Independent Handover

MIV Moving Image VideoGraphy

MOS Mean Opinion Score

MPEG Moving Picture Experts Group

MPEG-TS MPEG-Transport Stream

MPLS Multiprotocol Label Switching

MPQM Moving Picture Quality Metric

ms milisegundos

MSE Mean Squared Error

MSV Máquinas de Soporte Vectorial

MSVR Máquinas de Soporte Vectorial de Regresión

MVC Multi-View Video Coding

MVD Multi-View Vídeo Depth

NGN Next Generation Networking

NQM Noise quality measure

NQoS Network Quality of Service

- NR** Not Reference
- NVFM** Normalization Video Fidelity Metric
- PC** Paired Comparisons
- PCA** Principal Component Analysis
- PDA** Personal Digital Assistant
- PING** Packet Internet Groper
- PNG** Portable Network Graphics
- PSNR** Peak Signal to Noise Ratio
- PSNR-HVS** Peak Signal to Noise Ratio - Human Visual System
- PSQA** Pseudo-Subjective Quality Assessment
- P2P** Peer-to-Peer
- QoE** Quality of Experience
- QoS** Quality of Service
- QS** quantization scale
- RAM** Random-Access Memory
- RBF** Radial Basis Function
- RN** Redes Neuronales
- RNR** Redes Neuronales de Regresión
- RNRUP** Redes Neuronales de Regresión basadas en Unidades Producto
- RR** Reducted Reference
- RSSI** Received Signal Strength Indicator
- RTSP** Real Time Streaming Protocol
- RTT** Tiempo de retardo Round-Trip
- RTP** Real-Time Trans- port Protocol
- SC** Stimulus Comparison
- SCTP** Stream Control Transmission Protocol

- SDTV** Standard Definition Television
- SNMP** Simple Network Management Protocol
- SS** Single Stimulus
- SSCQE** Single-Stimulus Continous-Quality Evaluation
- SSIM** Single-scale Structural SIMilarity
- STB** Set-Top-Box
- STD** Standard Deviation
- SVC** Stereo Video Coding
- SVM** Support Vector Machine
- TCP** Transmission Control Protocol
- TCP/IP** Transmission Control Protocol/Internet Protocol
- TTL** Time To Live
- UDP** User Datagram Protocol
- UIT** Unión Internacional de Telecomunicaciones
- UML** Unified Modeling Language
- UMTS** Universal Mobile Telecommunications System
- UQI** Universal Quality Index
- URL** Uniform Resource Locator
- VBR** Variable Bitrate
- VCD** Video CD
- VIF** Visual Information Fidelity
- VLC** VideoLAN media player
- VoD** Video on Demand
- VoIP** Voice on IP
- VPN** Virtual Private Network
- VQ** Video Quality

VQM Video Quality Metric

VSNR Visual Signal-to-Noise Ratio

WIFI Wireless Fidelity

WIMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Network

WMV Windows Media Video

WSNR Weighted Signal-to-Noise Ratio

2D+Z 2D-plus-Depth

3D-IPTV 3D-TV sobre IP

3D/ST 3D/estereoscópico

3DTV 3D Television

Referencias

- A. M. Tekalp, E. Kurutepe y M. R. Civanlar (2007). “3DTV over IP: End-to-end streaming of multi-view video”. En: *IEEE Signal Processing Magazine*, 24.5, págs. 77-87.
- Agboma, Florence y Antonio Liotta (2008). “QoE-aware QoS Management”. En: *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*. MoMM '08. Linz, Austria, 24-26 November. DOI: 10.1145/1497185.1497210.
- (2012). “Quality of experience management in mobile content delivery systems”. En: *Telecommunication Systems*, 49.1, págs. 85-98. DOI: 10.1007/s11235-010-9355-6.
- Agrawal, D. y col. (2007). “A Scalable Approach for Supporting Streaming Media: Design, Implementation and Experiments”. En: *12th IEEE Symposium on Computers and Communications (ISCC)*. Aveiro, Portugal, July 1-4. DOI: 10.1109/ISCC.2007.4381589.
- Ahmad, A.M.A. (2008). “Multimedia Transcoding in Mobile and Wireless Networks”. En: Premier reference source. Information Science Reference.
- Alex J. Smola, Bernhard Schölkopf (2004). “A tutorial on support vector regression”. En: *Statistics and Computing*, 14.3, págs. 199-222. DOI: 10.1023/B:STCO.0000035301.49549.88.
- American National Standards Institute*. Available at <http://www.ansi.org>. Last Accessed: 2015-09-16.
- Analyzer ClearSight*. Available at <http://www.clearsightnet.com/resources.php>. Last Accessed: 2015-08-13.

- Arulampalam, Ganesh y Abdesselam Bouzerdoum (2003). "A generalized feedforward neural network architecture for classification and regression". En: *Neural Networks* 16.5-6, págs. 561-568.
- Atayero, A.A. y col. (2011). "Estimation of the Visual Quality of Video Streaming Under Desynchronization Conditions". En: *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2.12, págs. 304-318. DOI: 10.14569/IJACSA.2011.021201.
- Atenas, M. y col. (2010). "IPTV Transcoding to Avoid Network Congestion". En: *Sixth International Conference on Networking and Services (ICNS)*. Cancun, Mexico, March 7-13, 2010. DOI: 10.1109/ICNS.2010.13.
- Balasubramaniam, S. y J. Indulska (2004). "Vertical handover supporting pervasive computing in future wireless networks". En: *Computer Communications*, 27.8, págs. 708 -719. DOI: <http://dx.doi.org/10.1016/j.comcom.2003.10.010>.
- Banodkar, D. y col. (2008). "Multicast instant channel change in IPTV systems". En: *3rd International Conference on COMMunication System softWARE and MiddlewaRE (COMSWARE)*. Bangalore, India, 5-10 January, 2008. DOI: 10.1109/COMSWA.2008.4554442.
- Barry, M., A.T. Campbell y A. Veres (2001). "Distributed control algorithms for service differentiation in wireless packet networks". En: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*. Anchorage, Alaska, April 22-26, 2001. DOI: 10.1109/INFCOM.2001.916786.
- Benoit, Alexandre y col. (2008). "Quality assessment of stereoscopic images". En: *EURASIP journal on image and video processing* 2008, Article-ID. DOI: doi: 10.1155/2008/659024.
- Bingham, C., M. Godfrey y J.W. Tukey (1967). "Modern techniques of power spectrum estimation". En: *IEEE Transactions on Audio and Electroacoustics*, 15.2, págs. 56-66. DOI: 10.1109/TAU.1967.1161895.
- Bishop, Christopher M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
- Bjork, Niklas y Charilaos Christopoulos (1998). "Transcoder architectures for video coding". En: *IEEE Transactions on Consumer Electronics*, 44.1, págs. 88-98.

- Bosc, E. y col. (2011). "Towards a New Quality Metric for 3-D Synthesized View Assessment". En: *IEEE Journal of Selected Topics in Signal Processing*, 5.7, págs. 1332-1343. DOI: 10.1109/JSTSP.2011.2166245.
- Bri, D. y col. (2013). "An Intelligent System to Detect the Type of Devices Sending and Receiving Data in the Network." En: *Network Protocols and Algorithms*, 5.2, págs. 72-91.
- Brooks, P. y B. Hestnes (2010). "User measures of quality of experience: why being objective and quantitative is important". En: *IEEE Network*, 24.2, págs. 8-13. DOI: 10.1109/MNET.2010.5430138.
- Calyam, Prasad y col. (2004). "Performance Measurement and Analysis of H.323 Traffic". En: *Passive and Active Network Measurement*. Ed. por Chadi Barakat y Ian Pratt. Vol. 3015. Lecture Notes in Computer Science. Springer Berlin Heidelberg, págs. 137-146. DOI: 10.1007/978-3-540-24668-8_14.
- Canovas, A. y col. (2009). "Multicast TV over WLAN in a University Campus Network". En: *Fifth International Conference on Networking and Services (ICNS '09)*. Valencia, Spain, April 20-25, 2009. DOI: 10.1109/ICNS.2009.105.
- Canovas, A. y col. (2010a). "3D IPTV Study and Performance". En: *Second International Conference on Evolving Internet (INTERNET)*, Valencia, Spain, September 20-25, 2010. DOI: 10.1109/INTERNET.2010.19.
- Canovas, A. y col. (2012). "Vertical WLAN handover algorithm and protocol to improve the IPTV QoS of the end user". En: *IEEE International Conference on Communications (ICC)*, Ottawa, Canada, June 10-15, 2012. DOI: 10.1109/ICC.2012.6364623.
- Canovas, Alejandro y col. (2010b). "Improving IPTV QoE taking the suitable MPEG-2/MPEG-4 Quantizer based on jitter, delay and lost packets measurements". En: *International Journal on Advances in Telecommunications*, 3.3-4, págs. 129-139.
- Carlsson, C. y P. Walden (2007). "Mobile TV - To Live or Die by Content". En: *40th Annual Hawaii International Conference on System Sciences, (HICSS '07)*, Waikoloa, USA, Jan. 3-6, 2007. DOI: 10.1109/HICSS.2007.382.
- Chan, J. y A. Seneviratne (1999). "A practical user mobility prediction algorithm for supporting adaptive QoS in wireless networks". En: *IEEE International*

- Conference on Networks (ICON '99)*, Brisbane, Queensland, Australia, 28 Sept.-1 Oct, 1999. DOI: 10.1109/ICON.1999.796167.
- Chaudhry, A. y J.Y. Khan (2009). "An Efficient MBMS Content Delivery Scheme over the HSDPA Network". En: *4th International Symposium on Wireless Pervasive Computing (ISWPC)*, Melbourne, VIC, 11-13 Feb., 2009. DOI: 10.1109/ISWPC.2009.4800565.
- Chervenets, V. y col. (2016). "QoS/QoE correlation modified model for QoE evaluation on video service". En: *13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TC-SET'16)*. IEEE. Lviv, Ukraine, 23-26 Feb., 2016. DOI: 10.1109/TCSET.2016.7452146.
- Cisco (2007). *How Cisco Designed and Deployed IPTV for Global Communications*. Case Study. Available at http://www.cisco.com/web/about/ciscoitawork/downloads/ciscoitawork/pdf/cisco_it_case_study_ip_tv.pdf.
- D. Hoffman G. Fernando, V. Goyal y M. Civanlar (1998). *RFC 2250: RTP Payload Format for MPEG1/MPEG2 Video*. Inf. téc. United States: IETF.
- Dai, Z. y col. (2008). "Vertical Handover Criteria and Algorithm in IEEE802.11 and 802.16 Hybrid Networks". En: *IEEE International Conference on Communications (ICC '08)*, Beijing, China, 19-23 May., 2008. DOI: 10.1109/ICC.2008.470.
- Dekleva, Sasha y col. (2007). "Evolution and Emerging Issues in Mobile Wireless Networks". En: *Communications of the ACM*, 50.6, págs. 38-43. DOI: 10.1145/1247001.1247003.
- Dodgson, Neil A. (2005). "Autostereoscopic 3D Displays". En: *Computer*, 38.8, págs. 31-36. DOI: <http://doi.ieeecomputersociety.org/10.1109/MC.2005.252>.
- Drucker, Harris y col. (1997). "Support Vector Regression Machines". En: *Advances in neural information processing systems*, 9, págs. 155-161.
- Durbin, R y D Rumelhart (1989). "Product Units: A Computationally Powerful and Biologically Plausible Extension to Backpropagation Networks," en: *Neural Computation* 1.1, págs. 133-142. DOI: 10.1162/neco.1989.1.1.133.

- DVB worldwide*. Available at <https://www.dvb.org/news/worldwide>. Last Accessed: 2015-08-13.
- El-Melegy, Moumen T. (2014). "Model-wise and Point-wise Random Sample Consensus for Robust Regression and Outlier Detection". En: *Neural Network*, 59, págs. 23-35. DOI: 10.1016/j.neunet.2014.06.010.
- El-Yamany, N.A. y col. (2010). "Evaluation of depth compression and view synthesis distortions in multiview-video-plus-depth coding systems". En: *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*. Tampere, Finland, June 7-9, 2010. DOI: 10.1109/3DTV.2010.5506511.
- ElecCard Video Quality Estimator*. Available at <http://www.elecard.com/products/products-pc/professional/video-quest/>. Last Accessed: 2015-08-13.
- ETSI/TR/102 643 (2010). *ETSI TR 102 643. Human Factors (HF); Quality of Experience (QoE) requirements for real-time communication services*. http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=33382.
- Evalvid*. Available at <http://www.tkn.tu-berlin.de/menue/research/evalvid/>. Last Accessed: 2015-08-16.
- Favalora, G.E. (2005). "Volumetric 3D Displays and Application Infrastructure". En: *Computer*, 38.8, págs. 37-44. DOI: 10.1109/MC.2005.276.
- Fecker, U. y A. Kaup (2004). *Transposed picture ordering for dynamic light field coding*. ISO/IEC JTC1/SC29/WG11 Doc M10929, Redmond Usa, July 2004.
- Fernández-Escribano, Gerardo y col. (2006). "Transcodificación Heterogénea de Vídeo MPEG-2/H. 264. Predicción Intra-Frame." En: *XVII Jornadas de Paralelismo (JP 2006)*. Albacete, España, 19 Aug., 2006.
- FFMPEG*. Available at <http://www.ffmpeg.org>. Last Accessed: 2015-08-13.
- Fodor, G., A. Eriksson y A. Tuoriniemi (2003). "Providing quality of service in always best connected networks". En: *IEEE Communications Magazine*, 41.7, págs. 154-163. DOI: 10.1109/MCOM.2003.1215652.
- Frank, P. y J. Incera (2006). "A Neural Network Based Test Bed for Evaluating the Quality of Video Streams in IP Networks". En: *Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Morelos, Mexico, 26-29 Sept., 2006. DOI: 10.1109/CERMA.2006.7.

- G. Florido F. Liberal, J. O. Fajardo. (2009). "QoS-oriented admission control in HSDPA networks". En: *Network Protocols and Algorithms (NPA)*, 1.1, págs. 52-61. DOI: 10.5296/npa.v1i1.178.
- García, M. y col. (2009). "A QoE Management System for Ubiquitous IPTV Devices". En: *Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '09)*. Sliema, Malta, October 11-16, 2009. DOI: 10.1109/UBICOMM.2009.31.
- Gross, J. y col. (2004). "Cross-layer optimization of OFDM transmission systems for MPEG-4 video streaming". En: *Computer Communications*, 27, págs. 1044-1055.
- Gupta, Priyanka y Ajay Dureja (2015). "A survey on Real-Time Traffic Monitoring & Dynamic Profile Management in Mobile Data Networks". En: *International Journal of Innovations & Advancement in Computer Science (IJACS)*, 4.1, págs. 654-660.
- Haykin, Simon (1994). *Neural Networks: A Comprehensive Foundation*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Hellberg, Chris, Dylan Greene y Truman Boyes (2007). *Broadband Network Architectures: Designing and Deploying Triple-Play Services*. Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Hervás-Martínez, Cesar y col. (2006). "Improving Microbial Growth Prediction by Product Unit Neural Networks". En: *Journal of Food Science*, 71.2, págs. M31-M38. DOI: 10.1111/j.1365-2621.2006.tb08904.x.
- Hewage, C.T.E.R. y col. (2007). "Stereoscopic TV over IP". En: *4th European Conference on Visual Media Production (IETCVMP)*. London, United Kingdom, Nov. 27-29, 2007.
- Hewage, C.T.E.R. y col. (2009). "Quality Evaluation of Color Plus Depth Map-Based Stereoscopic Video". En: *IEEE Journal of Selected Topics in Signal Processing*, 3.2, págs. 304-318. DOI: 10.1109/JSTSP.2009.2014805.
- Heynderickx, Ingrid y Ronald Kaptein (2009). "Perception of detail in 3D images". En: *Image Quality and System Performance VI*. San Jose, California, USA, 19-21 January, 2009. DOI: 10.1117/12.806186.

- Holma, Harri, Martin Kristensson y Jorma Kaikkonen (2010). "Multimedia Broadcast Multicast Service (MBMS)". En: *WCDMA for UMTS*. John Wiley y Sons, Ltd, págs. 409-429. DOI: 10.1002/9780470669501.ch14.
- Hornik, K., M. Stinchcombe y H. White (1989). "Multilayer Feedforward Networks Are Universal Approximators". En: *Neural Netw.* 2.5, págs. 359-366. DOI: 10.1016/0893-6080(89)90020-8.
- Horváth, Gábor (2003). "Neural Networks in Measurement Systems". En: *Advances in Learning Theory: Methods, Models and Applications*, págs. 375-402.
- Huber, Peter J. (1964). "Robust estimation of a location parameter". En: *Annals of Mathematical Statistics*, 35.1, págs. 73-101. DOI: 10.1214/aoms/1177703732.
- Huynh-Thu, Q. y M. Ghanbari (2008). "Scope of validity of PSNR in image/video quality assessment". En: *Electronics Letters*, 44.13, págs. 800-801. DOI: 10.1049/e1:20080522.
- Hwang, Jeng-Neng y col. (1994). "Regression modeling in back-propagation and projection pursuit learning". En: *IEEE Transactions on Neural Networks*, 5.3, págs. 342-353. ISSN: 1045-9227. DOI: 10.1109/72.286906.
- IEEE/STD/802.21 (2008). *Media Independent Handover Services, IEEE*. <https://standards.ieee.org/about/get/802/802.21.html>.
- Imran, K., M. Mellia y M. Meo (2007). "Measurements of Multicast Television over IP". En: *15th IEEE Workshop on Local Metropolitan Area Networks (LANMAN 2007)*. Hyatt Regency, NY, USA, 10-13 Jun., 2007. DOI: 10.1109/LANMAN.2007.4295994.
- inCode Advisors (2006). *White paper: "The Quad-Play – the First Wave of the Converged Services Evolution"*. Available at <http://www.incodeconsulting.com/wp-content/uploads/pdf/whitepapers/3GSMConvergence-Feb-2006.pdf>. Last Accessed: 2015-08-15.
- International Electrotechnical Commission*. Available at <http://www.iec.ch>. Last Accessed: 2015-09-15.
- International Organization Standardization*. Available at <http://www.iso.org/iso/home.html>. Last Accessed: 2015-09-15.
- International Telecommunications Society*. Available at <http://www.itsworld.org>. Last Accessed: 2015-09-16.

- IP Video Quality Alliance*. Available at <http://www.ipvqa.org>. Last Accessed: 2014-02-16.
- IPTV Focus Group*. Available at <http://www.itu.int/ITU-T/IPTV/>. Last Accessed: 2015-08-18.
- ITU-T/BT.500 (2012). *ITU-T Recomendación BT.500. Metodología para la evaluación subjetiva de la calidad de las imágenes de televisión*. <https://www.itu.int/rec/R-REC-BT.500/es>.
- ITU-T/E.800 (2008). *ITU-T Recommendation E.800. Quality of Telecommunication Services: Concepts, Models, Objectives and Dependability Planning. Terms and Definitions Related to the Quality of Telecommunication Services*. <http://www.itu.int/rec/T-REC-E.800/en>.
- ITU-T/E.802 (2007). *ITU-T Recomendación E.802. Marco y metodologías para la determinación y la aplicación de parámetros de calidad de servicio*. <https://www.itu.int/rec/T-REC-E.802-200702-I/es>.
- ITU-T/FG.IPTV (2007). *ITU-T Recommendation FG IPTV, Definition of Quality of Experience (QoE)*. <https://www.itu.int/md/T05-FG.IPTV-IL-0050/en>.
- ITU-T/FG/IPTV-DOC-0147 (2007). *Working Document: IPTV Services Requirements*. http://www.itu.int/md/T05-FG.IPTV-DOC-0147/_page.print.
- ITU-T/G.114 (2003). *ITU-T Recommendation G.114, One way transmission time*. <https://www.itu.int/rec/T-REC-G.114/es>.
- ITU-T/H.262 (2012). *ITU-T Recommendation H.262. Information technology - Generic coding of moving pictures and associated audio information: Video*. <http://www.itu.int/rec/T-REC-H.262>.
- ITU-T/J.143 (2000). *ITU-T Recommendation J.143. User requirements for objective perceptual video quality measurements in digital cable television*. <http://www.itu.int/rec/T-REC-J.143/en>.
- ITU-T/J.144 (2004). *ITU-T Recomendación J.144. Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*. <https://www.itu.int/rec/T-REC-J.144-200403-I/en>.
- ITU-T/P.800 (1996). *ITU-T Recomendación P.800. Métodos de determinación subjetiva de la calidad de transmisión*. <https://www.itu.int/rec/T-REC-P.800/es>.

- ITU-T/P.862 (2001). *ITU-T Recomendación P.862. Evaluación de la calidad vocal por percepción: Un método objetivo para la evaluación de la calidad vocal de extremo a extremo de redes telefónicas de banda estrecha y códecs vocales*. <https://www.itu.int/rec/T-REC-P.862/es>.
- ITU-T/P.910 (2008). *ITU-T Recomendación P.910. Métodos subjetivos de evaluación de la calidad vídeo para aplicaciones multimedios*. <https://www.itu.int/rec/T-REC-P.910/es>.
- JavaNNS*. Available at <http://www.ra.cs.uni-tuebingen.de/software/JavaNNS/>. Last Accessed: 2015-08-13.
- Jimenez, J. M. y col. (2015). “New algorithm to improve QoE of IPTV service customers”. En: *Communications Software, Services and Multimedia Applications Symposium (CSSMA)*. London, UK, 8-12 June, 2015.
- JMVC*. Available at <http://research.nokia.com/page/4988>. Last Accessed: 2015-08-13.
- Joachims, T. (2012). *Learning to Classify Text Using Support Vector Machines*. The Springer International Series in Engineering and Computer Science. Springer US.
- Joe, Inwheel, Won-Tae Kim y Seokjoon Hong (2007). “A Network Selection Algorithm considering Power Consumption in Hybrid Wireless Networks”. En: *16th International Conference on Computer Communications and Networks (ICCCN 2007)*. Honolulu, Hawaii, USA, August 13-16, 2007. DOI: 10.1109/ICCCN.2007.4317990.
- Joo, Hyunchul y col. (2008). “An Effective IPTV Channel Control Algorithm Considering Channel Zapping Time and Network Utilization”. En: *IEEE Transactions on Broadcasting*, 54.2, págs. 208-216. DOI: 10.1109/TBC.2008.915767.
- Joveluro, P. y col. (2010). “Perceptual Video Quality Metric for 3D video quality assessment”. En: *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Tampere, Finland, June 7-9, 2010. DOI: 10.1109/3DTV.2010.5506331.
- Juluri, P., V. Tamarapalli y D. Medhi (2016). “Measurement of Quality of Experience of Video-on-Demand Services: A Survey”. En: *IEEE Communications Surveys Tutorials*, 18.1, págs. 401-418. DOI: 10.1109/COMST.2015.2401424.

Kecman, V. (2001). *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. Bradford Book.

KEEL. Available at <http://www.keel.es>. Last Accessed: 2015-08-13.

Kim, Hyun Jong y Seong Gon Choi (2010). "A study on a QoS/QoE correlation model for QoE evaluation on IPTV service". En: *The 12th International Conference on Advanced Communication Technology (ICACT)*, Gangwon-Do, Korea (South), 7-10 Feb., 2010.

Kim, Hyun-Jong y col. (2008a). "The QoE Evaluation Method through the QoS-QoE Correlation Model". En: *Fourth International Conference on Networked Computing and Advanced Information Management (NCM '08)*. Vol. 2. Gyeongju, Korea (South), 2-4 Sept., 2008. DOI: 10.1109/NCM.2008.202.

Kim, Jinsul y col. (2008b). "IPTV Systems, Standards and Architectures: Part II - Heterogeneous Networks and Terminal-Aware QoS/QoE-Guaranteed Mobile IPTV Service". En: *IEEE Communications Magazine*, 46.5, págs. 110-117. DOI: 10.1109/MCOM.2008.4511658.

Kim, Jongryool y col. (2007). "Multi-View 3D Video Transport using Application Layer Multicast with View Switching Delay Constraints". En: *3DTV Conference, 2007*. Kos, Greece, 7-9 May., 2007. DOI: 10.1109/3DTV.2007.4379478.

Kim, Jongryool y col. (2008c). "Challenges and Realization of Real-Time Multi-view Video Transport Systems over IP Networks". En: *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '08)*. Harbin, China, 15-17 Aug., 2008. DOI: 10.1109/IIH-MSP.2008.302.

Kim, Sang Soo y col. (2009). "31.1: Invited Paper: World's First 240Hz TFT-LCD Technology for Full-HD LCD-TV and Its Application to 3D Display". En: *SID Symposium Digest of Technical Papers*, 40.1, págs. 424-427. DOI: 10.1889/1.3256805.

Kim, Young il y col. (2010). "3D image transmission technology for mobile IPTV". En: *International Conference on Information and Communication Technology Convergence (ICTC)*. Jeju, korea, 17-19 Nov., 2010. DOI: 10.1109/ICTC.2010.5674681.

Klaue, Jirka, Berthold Rathke y Adam Wolisz (2003). "EvalVid - A Framework for Video Transmission and Quality Evaluation". En: *13th International Conference on Modelling Techniques and Tools for Computer Performance Eva-*

- luation. Urbana, IL, USA, September 2-5, 2003. DOI: 10.1007/978-3-540-45232-4_16.
- Knightson, K., N. Morita y T. Towle (2005). "NGN architecture: generic principles, functional architecture, and implementation". En: *IEEE Communications Magazine*, 43.10, págs. 49-56. DOI: 10.1109/MCOM.2005.1522124.
- Kondrad, Lukasz, Imed Bouazizi y Moncef Gabbouj (2008). "Seamless Handover for Mobile TV over DVB-H Applications". En: *7th International Conference on Mobile and Ubiquitous Multimedia (MUM '08)*. Ume, Sweden, December 3-5, 2008. DOI: 10.1145/1543137.1543174.
- Kooij, Robert E, Ahmed Kamal y Kjell Brunnström (2006). "Perceived quality of channel zapping." En: *Communication Systems and Networks*. Palma de Mallorca, Spain, August 28-30, 2006.
- Kosko, Bart (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- Kuran, Mehmet S. y Tuna Tugcu (2007). "A survey on emerging broadband wireless access technologies". En: *Computer Networks*, 51.11, págs. 3013 -3046. DOI: <http://dx.doi.org/10.1016/j.comnet.2006.12.009>.
- Lambooi, Marc T. M., Wijnand A. IJsselsteijn e Ingrid Heynderickx (2007). "Visual discomfort in stereoscopic displays: a review". En: *Stereoscopic Displays and Virtual Reality Systems XIV*, San Jose, California, USA, January 29-31, 2007. DOI: 10.1117/12.705527.
- Latré, Steven y col. (2009). "An Autonomic Architecture for Optimizing QoE in Multimedia Access Networks". En: *Comput. Netw.* 53.10, págs. 1587-1602. DOI: 10.1016/j.comnet.2008.11.004.
- Lee, Dai-Boong, Wan Kim y Hwangjun Song (2010). "An Effective Mobile IPTV Channel Control Algorithm over WiMAX Network". En: *7th IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, Nevada, USA, 9-12 Jan., 2010. DOI: 10.1109/CCNC.2010.5421648.
- Lee, Dai-Boong y Hwangjun Song (2010). "QoE-aware mobile IPTV channel control algorithm over WiMAX network". En: *Journal of Visual Communication and Image Representation*, 21.3, págs. 245 -255. DOI: <http://dx.doi.org/10.1016/j.jvcir.2010.01.002>.

- Lee, Kye-Hwan y col. (2008). “QoS-Guaranteed IPTV Service Provisioning in Home Network with IEEE 802.11e Wireless LAN”. En: *IEEE Network Operations and Management Symposium (NOMS)*, Salvador da Bahia, Brasil, 7-11 April, 2008. DOI: 10.1109/NOMSW.2007.15.
- Lee, Shie-Jue y Chun-Liang Hou (2002). “An ART-based construction of RBF networks”. En: *IEEE Transactions on Neural Networks*, 13.6, págs. 1308-1321. DOI: 10.1109/TNN.2002.804308.
- Li, Ming, Marcus Magnor y Hans-Peter Seidel (2003). “Online accelerated rendering of visual hulls in real scenes”. En: *Journal of WSCG*, 2.11, págs. 290-297.
- Liu, Yanwei y col. (2012). “QoE-oriented 3D Video Transcoding for Mobile Streaming”. En: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 8.3s, 42:1-42:20. DOI: 10.1145/2348816.2348821.
- Lloret, J. y col. (2011a). “A network management algorithm based on 3D coding techniques for stereoscopic IPTV delivery”. En: *IEEE Global Communications (GLOBECOM)*, Houston, TX, USA, 5-9 Dec., 2011. DOI: 10.1109/GLOCOMW.2011.6162447.
- Lloret, J. y col. (2011b). “A QoE management system to improve the IPTV network”. En: *International Journal of Communication Systems*, 24.1, págs. 118-138. DOI: 10.1002/dac.1145.
- Lloret, J. y col. (2011c). “A stereoscopic video transmission algorithm for an IPTV network based on empirical data”. En: *International Journal of Communication Systems*, 24.10, págs. 1298-1329. DOI: 10.1002/dac.1196.
- Lloret, J. y col. (2012). “A network management algorithm and protocol for improving QoE in mobile IPTV”. En: *Computer Communications*, 35.15, págs. 1855-1870. DOI: <http://dx.doi.org/10.1016/j.comcom.2012.06.002>.
- Lloret, J. y col. (2013). “A network algorithm for 3D/2D IPTV distribution using WiMAX and WLAN technologies”. English. En: *Multimedia Tools and Applications* 67.1, págs. 7-30. DOI: 10.1007/s11042-011-0929-4.
- Lopez, D. y col. (2006). “Adaptive multimedia streaming over IP based on customer oriented metrics”. En: *International Symposium on Computer Networks*, Istanbul, Turkey, 16-18 June, 2006. DOI: 10.1109/ISCN.2006.1662531.

- Lopez-Herreros, A. y col. (2015). "A New IP Video Delivery System for Heterogeneous Networks using HTML5". En: *IEEE International Conference on Communications (ICC)*, London, United Kingdom, June 8-12, 2015.
- Ma, Li y col. (2004). "A new method to support UMTS/WLAN vertical handover using SCTP". En: *IEEE Wireless Communications*, 11.4, págs. 44-51. DOI: 10.1109/MWC.2004.1325890.
- Matusik, Wojciech y Hanspeter Pfister (2004). "3D TV: A Scalable System for Real-time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes". En: *ACM Trans. Graph.*, 23.3, págs. 814-824. DOI: 10.1145/1015706.1015805.
- McKeown, Nick y col. (2008). "OpenFlow: Enabling Innovation in Campus Networks". En: *SIGCOMM Comput. Commun. Rev.*, 38.2, págs. 69-74. DOI: 10.1145/1355734.1355746.
- Meesters, L.M.J., W.A. IJsselsteijn y P.J.H. Seuntjens (2004). "A survey of perceptual evaluations and requirements of three-dimensional TV". En: *IEEE Transactions on Circuits and Systems for Video Technology*, 14.3, págs. 381-391. DOI: 10.1109/TCSVT.2004.823398.
- Meyer, Ulrike (2006). "Secure Roaming and Handover Procedures in Wireless Access Networks". Tesis doct. TU Darmstadt.
- Ministerio de Industria, Turismo y comercio*. Available at <http://www.itsworld.org>. Last Accessed: 2015-09-15.
- Mobile 3DTV Solideyesight*. Available at <http://sp.cs.tut.fi/mobile3dtv/video-plus-depth>. Last Accessed: 2015-08-13.
- Mohamed, S. y G. Rubino (2002). "A study of real-time packet video quality using random neural networks". En: *IEEE Transactions on Circuits and Systems for Video Technology*, 12.12, págs. 1071-1083. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2002.806808.
- Morton, D.H. (1988). *3d video transmission*. Available at <https://www.google.com/patents/EP0267000A3?cl=es>. EP Patent App. EP19,870,309,703.
- Moving Picture Expert Group*. Available at <http://mpeg.chiariglione.org>. Last Accessed: 2015-09-15.

National Telecommunications and Information Administration. Available at <http://www.ntia.doc.gov>. Last Accessed: 2015-09-15.

NetDisturb. Available at <http://www.zti-telecom.com/brochuresN/NetDisturb%20Literature.pdf>. Last Accessed: 2015-08-13.

Ng, Ping Chung, Soung Chang Liew y Chinlon Lin (2005). "Voice over wireless LAN via IEEE 802.16 wireless MAN and IEEE 802.11 wireless distribution system". En: *International Conference on Wireless Networks, Communications and Mobile Computing*, Maui, HI, USA, 13-16 Jun., 2005. DOI: 10.1109/WIRLES.2005.1549460.

Nishimoto, Yusei, H. Imaizumi y N. Mita (2009). "Integrated Digital Rights Management for Mobile IPTV Using Broadcasting and Communications". En: *IEEE Transactions on Broadcasting*, 55.2, págs. 419-424. DOI: 10.1109/TBC.2009.2016496.

O'Driscoll, G. (2008). *Next Generation IPTV Services and Technologies*. Wiley.

Oh, Kwan-Jung y col. (2007). "Multi-View Video and Multi-Channel Audio Broadcasting System". En: *3DTV Conference*, Kos, Greece, 7-9 May, 2007. DOI: 10.1109/3DTV.2007.4379437.

Ohmori, S., Y. Yamao y N. Nakajima (2000). "The future generations of mobile communications based on broadband access technologies". En: *IEEE Communications Magazine*, 38.12, págs. 134-142. DOI: 10.1109/35.888267.

Open Networking Foundation (2012). *White paper: "Software-defined Networking: The New Norm for Networks"*. Available at <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>. Last Accessed: 2015-08-15.

Park, A.H. y J.K. Choi (2007). "QoS guaranteed IPTV service over Wireless Broadband network". En: *The 9th International Conference on Advanced Communication Technology*, Gangwon-Do, Korea (South), 12-14 Feb., 2007. DOI: 10.1109/ICACT.2007.358545.

Perkis, A., S. Munkeby y Odd Inge Hillestad (2006). "A model for measuring Quality of Experience". En: *7th Nordic Signal Processing Symposium (NORSIG)*, Reykjavik, Iceland, 7-9 Jun., 2006. DOI: 10.1109/NORSIG.2006.275209.

Petrovic, Goran, Dirk Farin y Peter H. N. de With (2008). "Toward 3D-IPTV: design and implementation of a stereoscopic and multiple-perspective video

- streaming system". En: *Proceedings of SPIE-IS&T Electronic Imaging*, vol. 6803. San Jose, California, USA, págs. 680310-680310-8, 2008. DOI: 10.1117/12.766327.
- Philips 3D Solutions*. Available at <http://www.business-sites.philips.com/3dsolutions/home/index.page>. Last Accessed: 2015-08-13.
- Pinson, M.H. y S. Wolf (2004). "A new standardized method for objectively measuring video quality". En: *IEEE Transactions on Broadcasting*, 50.3, págs. 312-322. ISSN: 0018-9316. DOI: 10.1109/TBC.2004.834028.
- Pontes, A. y col. (2008). "Handover management in integrated WLAN and mobile WiMAX networks". En: *IEEE Wireless Communications*, 15.5, págs. 86-95. DOI: 10.1109/MWC.2008.4653137.
- Poon, Wing-Fai, Kwok-Tung Lo y Jian Feng (2005). "Performance study for streaming layered encoded videos in broadcast environment". En: *Third International Conference on Information Technology and Applications (ICITA)*, vol. 2. Sydney, NSW, 4-7 July, 2005. DOI: 10.1109/ICITA.2005.215.
- Project-Team DIONYSOS*. Available at <https://raweb.inria.fr/rapportsactivite/RA2014/dionysos/uid14.html>. Last Accessed: 2015-09-15.
- Pyykkö, S. Jumisko y col. (2006). "Acceptance of audiovisual quality in erroneous television sequences over a DVB-H channel". En: *Proc. of Workshop in Video Processing and Quality Metrics for Consumer Electronics*, págs. 1-5, 2006.
- Rahrer, Tim, Riccardo Fiandra, Steven Wright y col. (2006). "Triple-play services quality of experience (QoE) requirements". En: *DSL-Forum. Technical Report TR-126*.
- Rong, Bo y col. (2009). "Improving Delay and Jitter Performance in Wireless Mesh Networks for Mobile IPTV Services". En: *IEEE Transactions on Broadcasting*, 55.3, págs. 642-651. DOI: 10.1109/TBC.2009.2027739.
- S. Winkler (2007). *Symmetricom white paper: An Important Measure of Success for IP-based Video Services*. Available at http://qoe.symmetricom.com/pdf/WP_QoE_IPVS.pdf. Last Accessed: 2015-09-15.
- Saleem, L.N. y S. Mohan (2007). "An analysis of IP Multimedia Subsystems (IMS)". En: *First International Symposium on Advanced Networks and Telecommunication Systems*, New Delhi, India, 17-18 Dec., 2007. DOI: 10.1109/ANTS.2007.4620219.

- Saygili, G., C.G. Gürler y A.M. Tekalp (2009). "3D display dependent quality evaluation and rate allocation using scalable video coding". En: *16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, 7-10 Nov., 2009. DOI: 10.1109/ICIP.2009.5414317.
- Schollmeier, G. y C. Winkler (2004). "Providing sustainable QoS in next-generation networks". En: *IEEE Communications Magazine*, 42.6, págs. 102-107. DOI: 10.1109/MCOM.2004.1304243.
- Schulzrinne, H., A. Rao y R. Lanphier (1998). *RFC 2326: Real Time Streaming Protocol (RTSP)*. Inf. téc. United States: IETF.
- Schulzrinne, H. y col. (2003). *RFC 3550: RTP: A Transport Protocol for Real-Time Applications*. Inf. téc. IETF.
- Seuntiëns, P. (2006). "Visual experience of 3D TV". Tesis doct. Eindhoven University of Technology.
- Shi, Minghui, Xuemin Shen y Jon W. Mark (2004). "IEEE 802.11 roaming and authentication in wireless LAN/cellular mobile networks". En: *IEEE Wireless Communications*, 11.4, págs. 66-75. DOI: 10.1109/MWC.2004.1325893.
- Siller, M. (2006). "An agent-based platform to map quality of service to experience in active and conventional networks". Tesis doct. University of Essex.
- Siller, Mario y John Woods (2003). "Improving Quality of Experience for Multimedia Services by QoS arbitration on QoE Framework". En: *Proceedings of 13th Packed Video Workshop*. Nantes, France, April, 2003.
- Silverman, Peter y Michael Hanrahan (2012). *Broadband forum, Technical Report MR-180, "Achieving Quality IPTV over DSL"*. Inf. téc.
- Simoens, P. y col. (2008). "Design of an Autonomic QoE Reasoner for Improving Access Network Performance". En: *Fourth International Conference on Autonomic and Autonomous Systems (ICAS)*, Gosier, Guadeloupe, March 16-21, 2008. DOI: 10.1109/ICAS.2008.14.
- SNNS. Available at <http://www.ra.cs.uni-tuebingen.de/SNNS/>. Last Accessed: 2015-09-15.
- Soldani, D., M. Li y R. Cuny (2007). *QoS and QoE Management in UMTS Cellular Systems*. Wiley.

- Song, Young-Ho y Taeck-Geun Kwon (2009). "Fast Channel Change IPTV System for Enhanced User Experience". En: *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '09)*, Kyoto, Japan, 12-14 Sept., 2009. DOI: 10.1109/IIH-MSP.2009.12.
- Stojanovic, Mirjana D y col. (2015). "Approaches to Quality of Experience management in the future Internet". En: *12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS '15)*. IEEE. University of Nis, Serbia, 14-17 October, 2015. DOI: 10.1109/TELSIKS.2015.7357788.
- SVMLight*. Available at <http://svmlight.joachims.org>. Last Accessed: 2015-08-13.
- Telecommunication Networks Group*. Available at <http://www.tkn.tu-berlin.de/menue/research/evalvid/>. Last Accessed: 2015-09-15.
- Teyeb, Oumer y col. (2006). "Subjective Evaluation of Packet Service Performance in UMTS and Heterogeneous Networks". En: *Proceedings of the 2nd ACM International Workshop on Quality of Service & Security for Wireless and Mobile Networks*. Q2SWinet '06. Terromolinos, Spain, October 2-6, 2006. DOI: 10.1145/1163673.1163692.
- Total Video converter*. Available at <http://www.4easysoft.com/total-video-converter.html>. Last Accessed: 2015-08-13.
- Transition networks (2003). *White paper: "Quality of Service (QoS) in High-Priority Applications"*. Available at http://www.transition.com/TransitionNetworks/Uploads/Literature/qos_wp.pdf. Last Accessed: 2015-08-13.
- Unión Internacional de Telecomunicaciones*. Available at <http://www.itu.int/es/pages/default.aspx>. Last Accessed: 2015-09-15.
- Vanajakshi, L. y L.R. Rilett (2004). "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed". En: *IEEE Intelligent Vehicles Symposium*, Parma, Italy, 14-17 June, 2004. DOI: 10.1109/IVS.2004.1336380.
- Vapnik, Vladimir N. (1995). *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc.

- Vetro, A. y col. (2004). "Coding Approaches for End-to-End 3D TV Systems". En: *Proceedings of the Picture Coding Symposium (PCS)*. San Francisco, CA, Dec. 2004.
- Vetro, Anthony, Thomas Wiegand y Gary J Sullivan (2011). "Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard". En: *Proceedings of the IEEE*, 99.4, págs. 626-642.
- VLC*. Available at <http://www.videolan.org/vlc/>. Last Accessed: 2015-08-13.
- Wang, Kun (2013). *Thesis: "Stereoscopic 3D Video Quality of Experience: impact of coding, transmission and display technologies"*, Mid Sweden University.
- Wang, Yubing (2006). *Survey: "Survey of objective video quality measurements"*, Worcester Polytechnic Institute, USA.
- Watanabe, K. y col. (2008). "Proposal of new QoE assessment approach for quality management of IPTV services". En: *15th IEEE International Conference on Image Processing (ICIP)*, San Diego, CA, USA, 12-15 Oct., 2008. DOI: 10.1109/ICIP.2008.4712191.
- Welch, J. y J. Clark (2006). *RFC 4445: A Proposed Media Delivery Index (MDI)*. Inf. téc. 4445. IETF.
- Wiegand, Thomas y Gary J Sullivan (2007). "The H. 264/AVC video coding standard". En: *IEEE Signal Processing Magazine*, 24.2, págs. 148-153.
- Wireshark*. Available at <https://www.wireshark.org/download.html>. Last Accessed: 2015-08-13.
- Yaici, Karim y col. (2005). "User-centric Quality of Service Management in UMTS". En: *4th Latin American Network Operations and Management Symposium (LANOMS)*, August 29-31, 2005.
- Yamamoto, L.A.R. y J.G. Beerends (1997). "Impact of Network Performance Parameters on the End-to-End Perceived Speech Quality". En: *Proceedings of EXPERT ATM Traffic Symposium*, pág. 1997.
- Yamgo TV*. Available at <http://yamgo.com/>. Last Accessed: 2015-08-18.
- Yan, Xiaohuan, Y. Ahmet Şekercioglu y Sathya Narayanan (2010). "A Survey of Vertical Handover Decision Algorithms in Fourth Generation Heterogeneous

- Wireless Networks”. En: *Comput. Netw.*, 54.11, págs. 1848-1863. DOI: 10.1016/j.comnet.2010.02.006.
- Zaidi, M. y col. (2011). “Reducing Wi-Fi handover delay using a new positioning process”. En: *International Conference on Communications, Computing and Control Applications (CCCA)*, Hammamet, Tunisia, 3-5 Mar., 2011. DOI: 10.1109/CCCA.2011.6031469.
- Zeng, Qing-An y Dharma P. Agrawal (2002). “Handoff in Wireless Mobile Networks”. En: *Handbook of Wireless Networks and Mobile Computing*. John Wiley y Sons, Inc., págs. 1-25. DOI: 10.1002/0471224561.ch1.
- Zhang, Jin-Yu y Man-Gui Liang (2008). “IPTV QoS Implement Mechanism in WLAN”. En: *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '08)*, Harbin, China, 15-17 Aug., 2008. DOI: 10.1109/IIH-MSP.2008.166.