



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO DE FINAL DE CARRERA

TÍTULO DEL TFC: Sistema de Transmisión Streaming multidispositivo con aplicación HTML5 y Android.

TITULACIÓN: Ingeniería Técnica de Telecomunicación, especialidad Sistemas de Telecomunicación.

AUTORES: Celia Barca García
Juan Antonio Casado Henarejos

DIRECTOR / TUTOR: Francesc Tarrés Ruiz / Lluís Torres Urgell

FECHA: 5 de diciembre de 2012

***“El secreto de un gran negocio consiste
en saber algo más que nadie sabe.”***

(Aristóteles Onassis)

Título: Sistema de Transmisión Streaming multidispositivo con aplicación para HTML5 y Android

Autores: Celia Barca García, Juan Antonio Casado Henarejos

Director: Francesc Tarrés Ruiz, Lluís Torres Urgell

Fecha: 5 de diciembre de 2012

Resumen

El objetivo principal del proyecto del que es objeto el presente documento, es mostrar los pasos a seguir, así como detallar los elementos y tecnologías necesarias para proporcionar un servicio completo de distribución de contenidos audiovisuales multidispositivo por Internet de una entrada cualquiera de vídeo.

Debido al elevado coste económico que supone la implementación de un sistema comercial de estas características, se opta finalmente por adoptar una solución de coste cero con una arquitectura más modesta, basada en Software libre y gratuito, que permite comprobar y mostrar su funcionamiento completo de una forma sencilla e ilustrativa.

Sin embargo, y con el fin de no limitar la funcionalidad de este proyecto a un mero uso didáctico o casero, se decide no perder la referencia aportada por el punto de vista comercial a lo largo del proyecto, estudiando de manera teórica varias formas de aportar funcionalidades adicionales al sistema,

Dichas funcionalidades tienen como objetivo proporcionar al sistema una mayor flexibilidad y adecuación a las necesidades del mercado y del cliente, aportando una serie de ventajas consideradas lo suficientemente útiles y relevantes como para ser dignas de mención.

Este documento abarcará el proyecto completo, desde la oferta inicial hasta las aplicaciones necesarias para poder visualizar el contenido que utilizarán los usuarios finales, tanto desde ordenadores como desde dispositivos móviles Android.

Title: Multidevice Streaming Transmission's System with HTML5 and Android Application.

Author: Celia Barca García, Juan Antonio Casado Henarejos

Director: Francesc Tarrés Ruiz, Lluís Torres Urgell

Date: December, 5th 2012

Overview

The main objective of this project is to show the steps to follow and to reflect the detailed elements and technologies required to implement a full service of Internet distribution for an audiovisual multidevice of any video input.

Because of the high costs involved in the implementation of a commercial system of these characteristic, we decide to adopt a zero cost solution by using a simpler architecture of the system based on software which is both free and costless. This system allows us to demonstrate and prove its whole functioning in an easy illustrative form.

However, in order not to limit the alternative technologies services presented here to a mere theorization of its elements, or to home use, its also given the commercial point of view throughout this project, investigating theoretically the various ways in which to provide additional functionalities to the system.

These functionalities have the objective to provide the system with more flexibility and adequacy for the market and clients' needs, giving a number of advantages that have been considered useful and relevant enough to be worth mentioning.

This document encompasses the complete project, from the initial offer to the necessary applications which will enable the final user/s watch the entire content from computers and Android mobile devices.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. TECNOLOGÍA STREAMING.....	2
1.1. Historia	2
1.2. Definición	2
1.3. Características	4
1.4. Funcionamiento.....	4
1.5. Servicios.....	5
1.5.1. Streaming Live.....	5
1.5.2. Streaming Bajo Demanda	6
CAPÍTULO 2. ARQUITECTURA DEL SISTEMA	8
2.1. Objetivo	8
2.2. Descripción del sistema básico.....	8
2.3. Funcionalidades	9
2.4. Requerimientos.....	10
2.5. Posibles ampliaciones y mejoras	10
2.5.1. Codificadores.....	11
2.5.2. Servidores balanceados	13
2.5.3. CDN	14
CAPÍTULO 3. CODIFICACIÓN.....	16
3.1. Introducción.....	16
3.2. Escalabilidad	17
3.2.1. Escalabilidad de calidad	19
3.2.2. Escalabilidad espacial.....	19
3.2.3. Escalabilidad temporal.....	20
3.3. Parámetros de codificación	20
3.3.1. Parámetros de vídeo.....	21
3.3.2. Parámetros de audio.....	21
CAPÍTULO 4. COMPRESIÓN.....	23
4.1. Introducción.....	23
4.2. Tipos de compresión	23
4.2.1. Compresión sin pérdida	23

4.2.2.	Compresión con pérdida	24
4.3.	Compresión de audio.....	24
4.3.1.	Compresores de audio.....	25
4.4.	Compresión de vídeo.....	27
4.4.1.	Compresores de vídeo.....	30
CAPÍTULO 5. ENCAPSULAMIENTO Y TRANSPORTE		36
5.1.	Introducción.....	36
5.1.1.	MP4.....	36
5.1.2.	MPEG-TS.....	36
5.2.	Protocolos de transporte.....	39
5.2.1.	UDP.....	40
5.2.2.	RTP.....	40
5.2.3.	RTSP.....	41
CAPÍTULO 6. TRANSMISIÓN STREAMING.....		43
6.1.	Introducción.....	43
6.2.	Soluciones de mercado.....	43
6.3.	VLC Media Player	44
6.3.1.	Configuración de emisión	45
6.3.2.	Configuración de recepción	47
CAPÍTULO 7. SERVIDOR		49
7.1.	Introducción.....	49
7.2.	Características.....	49
7.3.	Servidor web.....	50
7.3.1.	Servicio APACHE.....	51
7.3.2.	Servicio FTP.....	52
7.4.	Webmin.....	53
7.5.	Firewall	54
CAPÍTULO 8. HTML5		56
8.1.	Introducción.....	56
8.2.	Características.....	56
8.2.1.	Marcado HTML	57
8.2.2.	Etiquetas básicas	58
8.2.3.	Hojas de estilos en cascada o CSS.....	59
8.3.	Estándar HTML5.....	59
8.4.	Desarrollo.....	60

CAPÍTULO 9. ANDROID	61
9.1. Introducción.....	61
9.2. Historia	61
9.3. Características.....	62
9.4. Actualizaciones y versiones	64
9.5. Desarrollo.....	66
CAPÍTULO 10. PRESUPUESTO	67
10.1. Introducción.....	67
10.2. Presupuesto del sistema de pruebas.....	67
CAPÍTULO 11. CONCLUSIONES	69
BIBLIOGRAFÍA	71

INTRODUCCIÓN

Hasta hace aproximadamente una década y, debido a las limitaciones de las distintas tecnologías de los sistemas de comunicaciones, la reproducción de contenido multimedia a través de Internet implicaba tener que descargar completamente el archivo en el disco duro local. Este factor provocaba que la visualización de contenidos fuese una tarea lenta y pesada, ya que los ficheros a descargar tienden a tener un tamaño considerable.

Actualmente, y gracias a los nuevos avances de las tecnologías, puede simplificarse esta tarea con la denominada “transmisión mediante Streaming”, basada en un sistema que permite acceder a un archivo situado en un servidor de Internet, descargarlo progresivamente y almacenarlo en un buffer¹.

Este proceso se va realizando a la vez que se va reproduciendo la parte del archivo que ya se ha descargado y, una vez se ha visualizado esa parte, se borra el buffer de datos. De esta manera, se ofrece a los usuarios finales la posibilidad de reproducir el contenido deseado al mismo tiempo que se descarga, minimizando así el tiempo de espera.

El objetivo de este proyecto es mostrar el proceso necesario a seguir, así como detallar los elementos y tecnologías necesarias, para proporcionar un servicio completo de distribución de contenidos audiovisuales multidispositivo por Internet de una entrada cualquiera de vídeo.

Puesto que no es posible simular la ingesta de señal Live de un canal de televisión, ya que conllevaría un coste económico muy alto, se comprobará el correcto funcionamiento del proyecto utilizando la señal de una Webcam de ordenador como señal Live.

Este documento abarcará el proyecto completo, desde la oferta inicial, contemplada en el capítulo 10 del presente documento, hasta las aplicaciones necesarias para poder visualizar el contenido que utilizarán los usuarios finales, tanto desde ordenadores como desde dispositivos móviles Android.

Además, incluirá explicaciones teóricas sobre algunas de las modificaciones y mejoras que podrían realizarse sobre este sistema para que se ajuste más a las necesidades del cliente, aportándole funcionalidades adicionales significativamente ventajosas que proporcionarían al sistema un valor y uso más comercial, sin necesidad de complicar excesivamente su arquitectura.

¹ Ubicación de la memoria en un disco reservada para el almacenamiento temporal de información digital, mientras espera a ser procesada.

CAPÍTULO 1. TECNOLOGÍA STREAMING

1.1. Historia

Antes de la aparición de la tecnología Streaming, la reproducción de contenido multimedia a través de Internet implicaba necesariamente tener que descargar de forma completa el archivo deseado en el disco duro local de la máquina para poder reproducirlo posteriormente.

Como el tamaño de los archivos de audio y vídeo tendía a ser muy grande, el hecho de descargarlos completamente se convertía en una operación muy lenta y complicada que exigía a los usuarios finales unos recursos disponibles y un tiempo de espera muy elevados.

Con el paso de los años, fue mejorándose e incrementándose la velocidad en las conexiones a Internet y, con ello, el tiempo de espera y los recursos necesarios fueron disminuyendo progresivamente, pero no se llegó a su máximo apogeo hasta la aparición de la tecnología Streaming como tal.

En 1995, surgió el primer protocolo de Streaming de la mano de RealNetworks², una compañía norteamericana proveedora de Software para Internet y servicios multimedia.

Inicialmente, solo fue creado para la transmisión de audio por Internet, bautizándolo como RealAudio®¹. Dos años más tarde, las innovaciones continuaron, llegando así la primera solución de Streaming para la transmisión de vídeo por Internet, llamada RealVideo®¹, que permitía acceder a contenidos audiovisuales sin necesidad de descarga previa.

1.2. Definición

La palabra “stream” proviene de la lengua inglesa y significa “flujo” o “corriente”. Este significado hace referencia a las transmisiones en directo que son, al fin y al cabo, una descarga de ficheros que no tiene principio ni fin.

Actualmente, existen tres tipos diferenciados de descarga:

- a. Descarga tradicional: los usuarios descargan los ficheros completos a su disco duro local y después los reproducen.

En el caso de la transferencia tradicional, el archivo de vídeo original es subido al servidor al completo. Una vez allí, éste envía el vídeo a una

² <http://www.realnetworks.com/our-story/>

Base de Datos donde se va almacenando progresivamente hasta alcanzar su totalidad. Una vez el vídeo completo ha sido almacenado en la Base de Datos, los usuarios finales pueden reproducirlo desde sus dispositivos, siguiendo el esquema que puede verse en la figura siguiente:

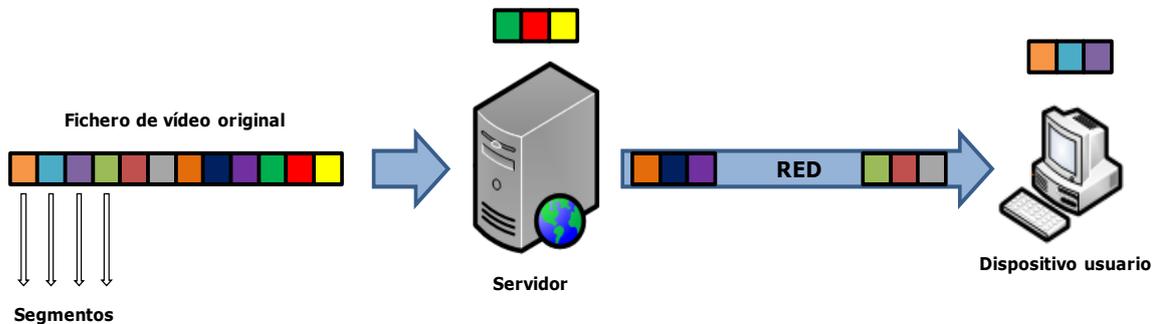


Fig. 1.1 Esquema de transferencia de vídeo tradicional

- b. Descarga progresiva: también denominada “pseudo-streaming”, donde los usuarios finales reproducen los archivos según los van descargando al disco o a la memoria de sus respectivos dispositivos.

Este tipo de descarga no puede darse con emisiones de contenido a tiempo real, ya que requiere que dicho contenido haya sido previamente grabado y almacenado.

- c. Streaming: los usuarios finales reproducen los archivos según les llegan segmentados a través de la Red y, una vez reproducidos, los descartan siguiendo los criterios que les marcan las políticas de caché que tienen definidas.

En el caso de la transferencia Streaming, el fichero de vídeo original es fragmentado en pequeños segmentos que se transfieren por la Red (flujo de paquetes) hasta llegar a los dispositivos de los usuarios finales. Estos pequeños paquetes se van reproduciendo según van llegando y van siendo descartados según han sido ya reproducidos.

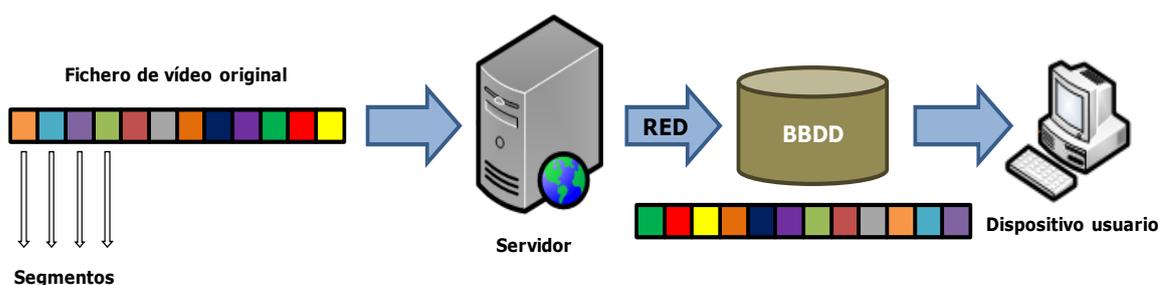


Fig. 1.2 Esquema de transferencia de vídeo Streaming

1.3. Características

El vídeo Streaming es una tecnología de transmisión de datos que se basa en un sistema que permite acceder a un archivo situado en un servidor de Internet, sin necesidad de descargarlo, y reproducir su contenido.

Para conseguirlo, los dispositivos de los usuarios finales almacenan el archivo en un buffer, cargando así antes una carpeta temporal con paquetes de datos en su memoria. Este proceso se va realizando a la vez que se va reproduciendo la parte del archivo que ya se ha descargado y, una vez se ha visualizado esa parte, se borra del buffer el flujo de datos.

La tecnología Streaming tiene varias peculiaridades, entre las que se destacan las siguientes:

- Los ficheros de Streaming pueden crearse en el momento de su difusión (Streaming Live) o estar almacenados previamente en un servidor (VoD³).
- Los usuarios finales pueden conectarse para visualizar el contenido en cualquier momento, no solo al principio de la transmisión.
- Los usuarios finales pueden visualizar los contenidos deseados sin la necesidad de esperar a que los ficheros se descarguen completamente.
- El cliente es capaz de entregar su contenido a los usuarios finales habiendo recibido éstos solo una pequeña parte de los archivos, mientras continúan recibiendo la corriente de datos (Streaming).
- La parte de los datos que se va almacenando puede actuar como seguro para las transmisiones con ancho de banda irregular: en caso de que la velocidad de transmisión de los ficheros sea menor que la velocidad a la que se está reproduciendo el contenido en momentos puntuales o durante ciertos intervalos de tiempo, se irán reproduciendo los datos almacenados de forma secuencial, con el objetivo de que los usuarios finales no sufran cortes ni interrupciones durante la visualización del contenido.

1.4. Funcionamiento

La tecnología Streaming se basa en el protocolo UDP/IP, detallado en el apartado 5.2.1 del presente documento, que permite la transmisión de contenido a través de la Red, asegurando una velocidad en la transmisión que

³ Sistema que permite al usuario el acceso a contenidos multimedia de forma personalizada ofreciéndole la posibilidad de solicitar y visualizar contenido concreto en el momento exacto en el que el usuario lo desee.

permite que los usuarios finales puedan visualizar el archivo seleccionado al mismo tiempo que se realiza la descarga.

El usuario final se conecta con el servidor donde están ubicados los contenidos multimedia con el fin de que éste empiece a enviarle el archivo deseado. El dispositivo del usuario final recibe la primera entrega de este fichero y va almacenando las entregas restantes en el buffer del reproductor multimedia para asegurar la fluidez de la emisión.

Cuando el buffer recibe una parte del archivo, el usuario final puede comenzar a reproducirlo a la vez que continúa descargándose. El sistema está sincronizado para que el vídeo pueda verse mientras continúa descargándose el archivo, de modo que cuando acaba la descarga del fichero, el usuario habrá finalizado, a su vez, la visualización del vídeo.

Este buffer sirve como “seguro” por si la conexión sufre descensos de velocidad o cortes en algún momento, permitiendo utilizar la información que ha sido almacenada en él hasta entonces para poder continuar con la reproducción del vídeo sin interrupciones (mecanismos de compensación), logrando así que esto no afecte a los usuarios finales.

Pero este sistema tiene un problema y es que, si la comunicación se interrumpe durante demasiado tiempo, el buffer se vaciará y la reproducción del vídeo se interrumpirá también hasta que se restaure la señal.

Por otra parte, si la velocidad de transmisión del Streaming es mayor que el ancho de banda del usuario final, éste consume por completo el contenido del buffer ya que se llena más despacio de lo que se está consumiendo, cosa que implica la detención de la emisión con el fin de poder cargar de nuevo el buffer.

1.5. Servicios

Dentro de la tecnología de Streaming existen dos tipos de servicio claramente diferenciados, en función de cómo se obtiene la información a difundir: en directo (Streaming Live) y Bajo Demanda.

1.5.1. Streaming Live

Cuando hablamos de Streaming Live o Streaming en directo, nos referimos a aquel que transmite eventos que están sucediendo justo en el momento de la difusión, como puede ser, por ejemplo, la señal “Live” que emiten los canales de televisión.

En este tipo de Streaming, se utiliza la denominada transmisión “broadcast” (difusión) porque realmente se está transmitiendo “en vivo” a todos los clientes

la misma información, que es el evento que se está produciendo en ese momento.

De esta manera, todos los usuarios finales ven exactamente el mismo punto del stream en un instante determinado independientemente de cuándo se conectan éstos al servidor, exceptuando las lógicas variaciones de los retardos en la red, que hacen que unos usuarios reciban antes los datos que otros.

Este concepto también es aplicable a la información que se difunde pero que no parte de un evento en directo, como puede ser el caso de contenido que se ha grabado previamente, pero que se va a difundir en un momento determinado (conocido como “transmisión en diferido”).

Para poder realizar este tipo de transmisión se necesita, entre otras cosas, un equipo que realice el proceso de captura y compresión en tiempo real (conocido como difusor o broadcaster) y un servidor de Streaming.

Si el número de usuarios que van a acceder al sistema es pequeño, este equipo puede estar instalado en la misma máquina que el servidor de Streaming. Sin embargo, si el número de usuarios es considerablemente grande, es recomendable tener, como mínimo, una máquina para realizar el proceso de captura y compresión en tiempo real y otra para el servidor de Streaming.

Hay que tener en cuenta que, para dar un servicio realmente eficiente de este tipo de Streaming, es conveniente que la difusión se realice con técnicas de multicast, de manera que el envío de información en la Red se realice a múltiples destinos de forma simultánea y bajo un único flujo de información.

1.5.2. Streaming Bajo Demanda

Cuando hablamos de servicio bajo demanda, más conocido como VoD⁴, nos referimos al que puede proporcionarnos, por ejemplo, un reproductor de vídeo, donde los usuarios finales acceden al contenido multimedia que desean visualizar mediante la realización de peticiones.

En este tipo de Streaming, la transmisión del medio empieza a ser reproducido para cada uno de los usuarios finales desde el inicio del evento y/o archivo. El fichero a transmitir puede estar ya preparado desde el comienzo del proceso en un archivo comprimido.

Normalmente, estos ficheros suelen estar ya situados en el punto de publicación correspondiente del servidor y preparados para la reproducción inmediata bajo petición de los usuarios.

⁴ Video On Demand (Vídeo Bajo Demanda). Sistema que permite al usuario el acceso a contenidos multimedia de forma personalizada ofreciéndole de este modo la posibilidad de solicitar y visualizar un vídeo en concreto en el momento en que el espectador lo desee.

Para facilitar la tarea de la identificación de los distintos contenidos audiovisuales, cada uno de ellos suele estar asociado a un thumbnail⁵ diferente, que es una versión reducida de una imagen, casi siempre extraída del mismo vídeo al que pretende describir.

Para este caso en particular y, a diferencia de para Streaming en directo, disponer de la posibilidad de realizar multicast en la red no representa ninguna ventaja adicional, ya que cada cliente recibe una parte distinta del stream y, por lo tanto, un paquete de datos diferente.

⁵ Los thumbails o miniaturas son versiones reducidas de imágenes, utilizadas para ayudar a la organización y reconocimiento de su respectivo contenido.

CAPÍTULO 2. ARQUITECTURA DEL SISTEMA

2.1. Objetivo

El objetivo de este sistema es proporcionar un servicio de distribución de contenidos audiovisuales por Internet de un canal cualquiera de televisión sobre ordenadores y dispositivos móviles.

Debido al elevado presupuesto que supondría su implementación real para un uso comercial, se utilizarán algunos elementos sustitutivos de uso libre y gratuito para validar el funcionamiento del sistema en sí.

Para la simulación de la señal Live que emitiría un canal de televisión, se utilizará una webcam de ordenador, que permitirá comprobar su funcionamiento con contenidos tanto Live como Bajo Demanda (VoD).

Puesto que es un sistema de transmisión mediante Streaming multidispositivo, es necesario un sistema de compresión, que reduzca el tamaño del archivo, minimizando así el tiempo de descarga, y de codificación, que convierta audio y vídeo a formatos diferentes en función del dispositivo desde el cual los usuarios finales realizarán la petición.

En lugar de comprar y programar los codificadores, se utilizará un Software gratuito que permitirá emularlos y conseguir así la codificación en diferentes formatos que nos proporcionarían éstos, pero de forma totalmente gratuita.

Finalmente, se incluirán en el diseño una serie de elementos que se describirán a nivel teórico a lo largo del presente capítulo, completamente opcionales, pero que aportan ventajas muy significativas a tener en cuenta, especialmente para la utilización del sistema a nivel comercial.

2.2. Descripción del sistema básico

El cliente transmite una señal de vídeo en directo (o en diferido) de su propio canal que pasa directamente al codificador del sistema, que se encarga de codificar esta señal con el objetivo de proporcionar un formato adecuado para la publicación de flujos de vídeo (streams) vía Internet.

Una vez el vídeo ha sido debidamente codificado, es transmitido y almacenado en el servidor web a través de un servicio FTP⁶, que se detallará en el apartado 7.3.2 del presente documento.

⁶ File Transfer Protocol (Protocolo de Tránsito de archivos).

La carpeta donde se van almacenando estos streams de vídeo es publicada por el servicio Apache, que se detallará en el apartado 7.3.1 del presente documento, de manera que la transferencia FTP ubica los streams de vídeo directamente en el punto de publicación.

Finalmente, los usuarios finales del servicio hacen la petición de vídeo al servidor web, recogiendo dichos flujos de su punto de publicación, y envía el contenido deseado al dispositivo del usuario final desde el cual se ha realizado la petición.

El diagrama de bloques del sistema básico puede verse a continuación:

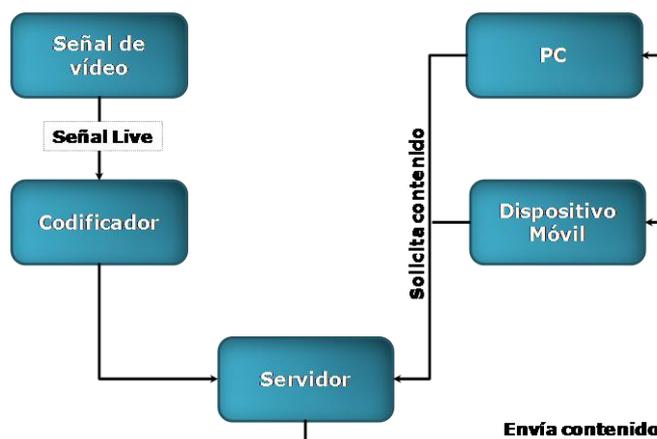


Fig. 2.1 Diagrama de bloques del sistema básico

Como se ha comentado anteriormente, este sistema y todas las variantes que se detallarán a lo largo del presente capítulo, están diseñados para poder ser implementados y utilizados a nivel comercial por un canal de televisión, aunque debido al coste económico que representarían dichos sistemas, algunos de los módulos solo se explicarán a nivel teórico.

2.3. Funcionalidades

Las funcionalidades previstas para este sistema son las siguientes:

- ❖ Transmisión de contenido multimedia tanto mediante Streaming Live como Streaming Bajo Demanda.
- ❖ Compresión del contenido multimedia de forma que permita transmisión por Streaming, tanto en directo como Bajo Demanda.
- ❖ Codificación del contenido multimedia de forma que permita la visualización de contenidos desde múltiples dispositivos.

- ❖ Publicación de contenidos multimedia en un servidor y acceso a ellos a través de una aplicación web.

2.4. Requerimientos

Debido a las limitaciones económicas explicadas a lo largo del presente documento, en este apartado se comentarán los requerimientos necesarios para comprobar el funcionamiento del sistema básico y no los que se requerirían para su implementación comercial:

- ❖ Un ordenador con webcam que simulará la emisión de la señal de vídeo de un canal de televisión y el servidor del cliente.
- ❖ Un software que simulará la codificación y compresión de contenido que proporcionaría el codificador del cliente.
- ❖ Una aplicación web desde la cual los usuarios finales accederán y se descargarán el contenido.
- ❖ Una aplicación para Android desde la cual los usuarios finales accederán y se descargarán el contenido.
- ❖ Un ordenador y un dispositivo móvil que actuarán como dispositivos de los usuarios finales.

2.5. Posibles ampliaciones y mejoras

Debido a la sencillez del diseño del sistema propuesto y a los avances de la tecnología, actualmente son posibles varias maneras diferentes de mejorar y ampliar dicho sistema, con el fin de aportar un mayor número de ventajas y/o de mejorar el nivel de adaptación a las necesidades del cliente.

Generalmente, la mayoría de clientes buscan la implementación de sistemas que puedan ofrecerles el servicio básico demandado que garantice ciertos niveles de calidad y al precio más bajo ya que, debido a limitaciones presupuestarias, no pueden permitirse sistemas demasiado sofisticados aunque éstos puedan aportarles funcionalidades extra.

Así pues, se presenta el esquema de arquitectura más sencillo para que el precio de la oferta realizada sea el más bajo posible y se plantean a nivel teórico las posibles mejoras, completamente opcionales, para su posterior estudio de viabilidad según el presupuesto.

2.5.1. Codificadores

Una de las posibles maneras de mejorar el sistema es utilizar varios codificadores en lugar de solo uno. A nivel de arquitectura, esto no aportaría una gran complejidad al sistema, como puede verse en la figura siguiente, ya que todos codificadores reciben exactamente la misma señal de vídeo Live emitida por el canal de televisión y la publican en su punto de publicación correspondiente dentro del servidor web.

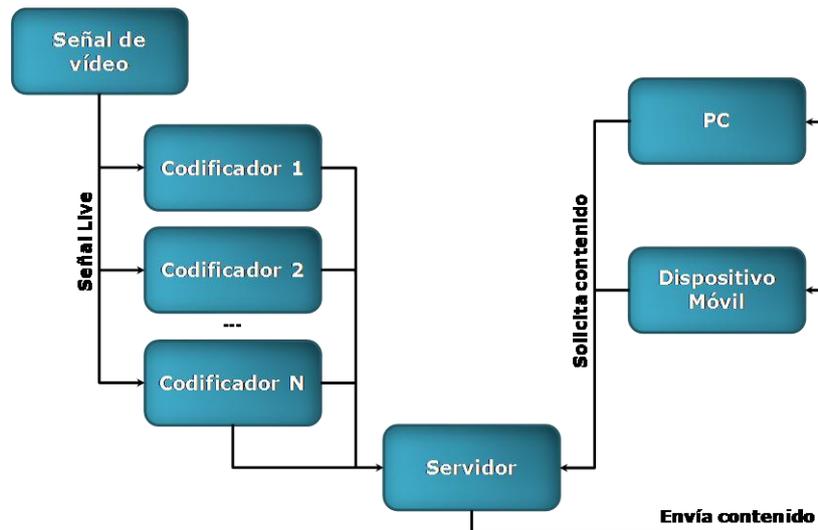


Fig. 2.2 Diagrama de bloques del sistema con N codificadores

Se plantean fundamentalmente tres opciones:

- a. Codificación en distintos formatos: la inclusión de uno o más codificadores que permitan codificar el vídeo ingestado en distintos formatos de forma que el contenido pueda ser accesible a través de un mayor número de tipos de dispositivos.

En este caso, los usuarios finales pueden escoger el codificador del cual van a visualizar los vídeos de forma manual a través del servidor web, escogiendo uno u otro punto de publicación, ya que cada codificador realizará su publicación en una carpeta diferente del servidor.

- b. Codificación escalable: la inclusión de uno o más codificadores que permitan la funcionalidad de escalabilidad, que se detallará en el apartado 3.2 del presente documento, adaptando así la calidad y resolución de los vídeos a la velocidad de transmisión de la red y al tipo de dispositivo de los usuarios.

En este caso, los usuarios podrían escoger la resolución y calidad de su elección de la misma manera que en el apartado anterior, aunque eso requeriría tener que codificar un mismo vídeo en múltiples resoluciones y calidades diferentes, que restarían eficiencia al sistema.

El uso de este tipo de codificadores está pensado más bien para la codificación en tiempo real (señal Live), de forma que el elemento realice la elección que más convenga al usuario de forma “inteligente”, en función de los recursos disponibles de éste.

- c. Codificación redundante: la inclusión de uno o más codificadores iguales al codificador propuesto en el sistema inicial. En este caso, la configuración de los codificadores adicionales es mucho más sencilla ya que simplemente hay que copiar la configuración del codificador principal.

Ambos codificadores recibirán exactamente la misma señal por parte del cliente y publicarán exactamente el mismo contenido en el servidor, aunque cada uno lo hará en su directorio correspondiente.

En este caso, dicha ampliación del sistema no aportará ventajas adicionales, sino que hará el sistema más robusto permitiendo así poder seguir visualizando contenidos aunque uno de los codificadores deje de funcionar correctamente, ya que podrá utilizarse el contenido codificado por uno de los codificadores redundantes.

El proceso de cambio de codificador principal a codificadores redundantes o viceversa, puede controlarse de forma manual a través del servidor web o de forma automática, monitorizando periódicamente el estado de los codificadores (por ejemplo, mediante “ping”) y programando un script para determinar el diagrama de estados que seguirá la conmutación de codificadores.

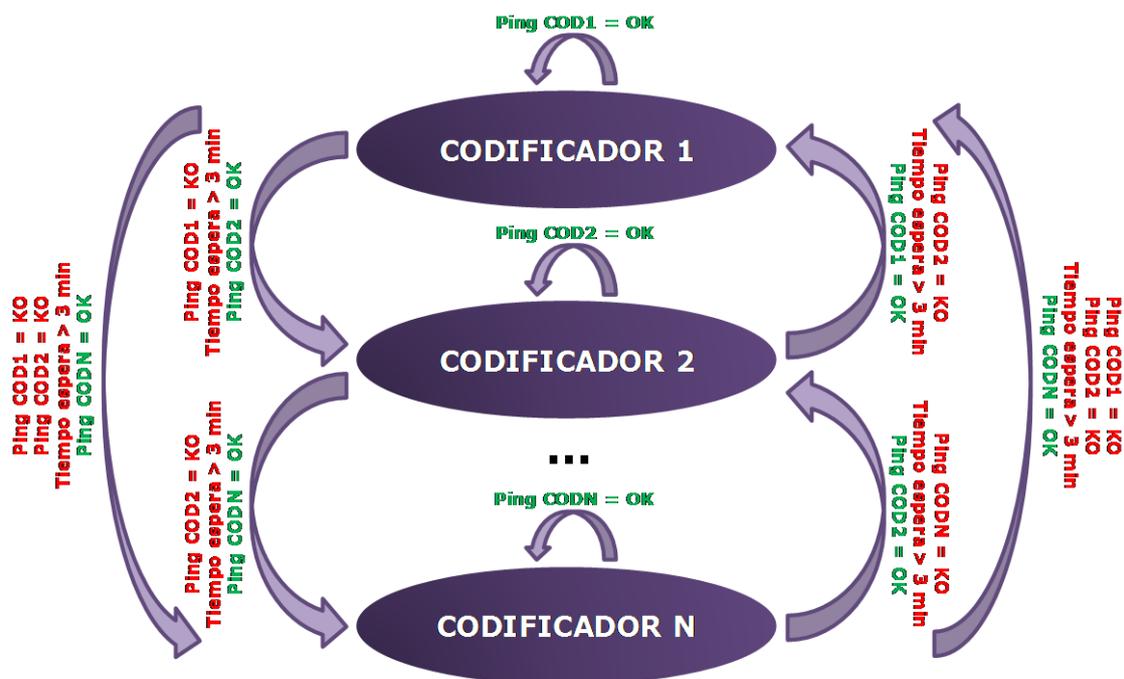


Fig. 2.3 Ejemplo de diagrama de estados con N codificadores

El sistema admite tantos codificadores y de tipos tan diversos como se quieran, incluyendo la combinación de estas opciones, dentro de los límites de compatibilidad de los elementos.

Como se puede ver, el hecho de añadir más codificadores implica obtener ventajas obvias frente al hecho de tener solo uno, aunque significa también que el cliente debe realizar una inversión mucho mayor.

2.5.2. Servidores balanceados

Otra de las formas de mejorar este sistema será mediante la implementación de dos o más servidores balanceados. A nivel de diseño de arquitectura, la inclusión de esta modificación será fácil de realizar, tal y como podemos ver en la figura siguiente, ya que solamente hay que incluir un balanceador antes de llegar a los servidores y tantos servidores como se deseen.

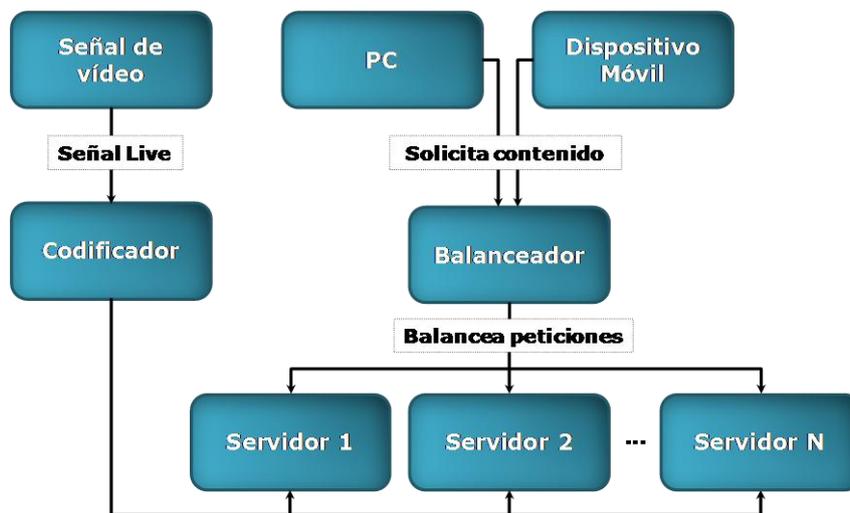


Fig. 2.4 Diagrama de bloques del sistema con servidores balanceados

Un balanceador es un dispositivo hardware que se pone al frente de, en este caso particular, un conjunto de servidores que atienden a una aplicación y asigna o balancea las solicitudes que llegan de los usuarios finales a los servidores utilizando un algoritmo.

Sin esta mejora y manteniendo el esquema básico del sistema, los usuarios finales harán sus peticiones a la dirección IP del servidor web donde están almacenados los contenidos que éstos desean visualizar. El servidor recibirá la petición y la responderá, proporcionando así los contenidos deseados.

Con la implementación de servidores balanceados, los usuarios finales harán peticiones a la dirección IP del balanceador, del que colgarán los N servidores web de los que disponga el sistema. Este balanceador mandará las peticiones

a un servidor o a otro dependiendo de la carga que estén soportando éstos y del algoritmo que utilice.

Esta modificación incrementa notablemente la capacidad de respuesta de las peticiones por parte de los servidores web ya que, de esta forma, se evita que los servidores lleguen a verse sobrecargados y no puedan proporcionar el servicio deseado a los usuarios finales, o que tengan unos tiempos de respuesta demasiado elevados como para ser aceptables.

2.5.3. CDN

Otra de las formas de mejorar este sistema es mediante la inclusión de una CDN⁷. Esta modificación en la arquitectura es, si cabe, más sencilla que las anteriores, tal y como podemos ver en la figura siguiente, debido a que estas Redes de Distribución de Contenidos suelen ser gestionadas por terceros, permitiendo así que el cliente pueda desentenderse de su creación, configuración y mantenimiento.

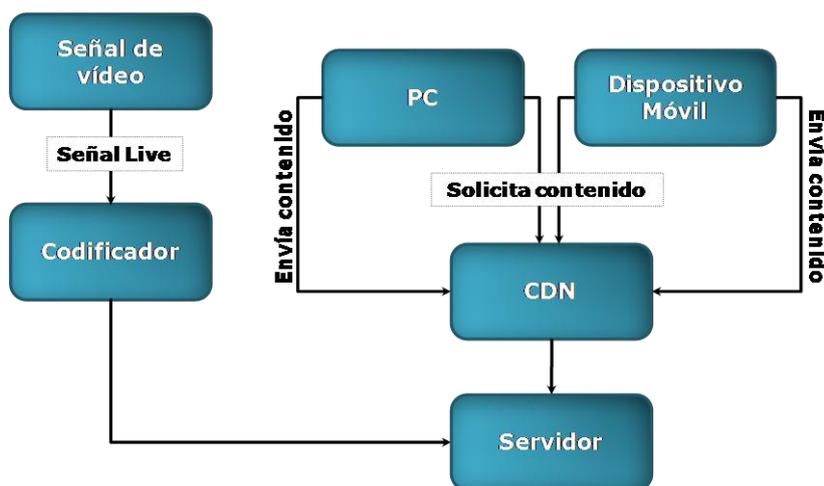


Fig. 2.5 Diagrama de bloques del sistema con CDN

Una CDN es un conjunto de servidores ubicados en diferentes puntos de una red que contienen copias locales de ciertos contenidos que están almacenados en otros servidores, de forma que éstos primeros se puedan servir de manera más eficiente, reduciendo de esta forma la latencia.

Esta mejora en la eficiencia se logra con un mejor balanceo en la carga a la que están sometidos, tanto los servidores que alojan los contenidos como los enlaces que interconectan las distintas secciones de la red, sirviendo los datos en función de la cercanía geográfica al usuario final.

⁷ Content Delivery Network (Red de Distribución de Contenidos).

De este modo se evita que algunos servidores se colapsen por exceso de peticiones, gracias a la distribución geográfica de los datos, y se minimizan los retardos, ya que el camino hasta el contenido es el mínimo posible.

Las CDNs nacen de la necesidad de optimizar este tipo de sistemas ya que minimizan de forma considerable el consumo de memoria y CPU y el espacio en disco de los servidores, permitiendo así que los requisitos de los servidores que conforman el sistema sean menores y, por tanto, también su precio.

Debido a sus características de funcionamiento, son especialmente recomendables para servidores de Streaming, ya que pueden servir los contenidos de forma más eficiente y rápida a los usuarios finales gracias a la distribución geográfica de los datos.

Además de las ya comentadas, el uso de estas Redes proporciona las ventajas siguientes:

- ❖ Reduce la carga de los servidores, así como la latencia y la demora y pérdida de paquetes.
- ❖ Proporciona una red de tráfico distribuida.
- ❖ Incrementa el ancho de banda y la capacidad de conexión.
- ❖ Garantiza el 100% de disponibilidad de información ante la caída de alguno de los servidores.

Permite obtener estadísticas de comportamiento de usuarios como, por ejemplo, qué páginas visitan o desde qué ubicación geográfica.



Fig. 2.6 Ejemplo de un mapa de Red de CDN Level3

CAPÍTULO 3. CODIFICACIÓN

3.1. Introducción

La codificación en Streaming nace un cúmulo de necesidades que se han ido generando conforme ha ido avanzando la tecnología con el objetivo de mejorar y hacer más eficientes estos sistemas.

Con el fin de adaptarse a las nuevas necesidades del mercado, las distintas tecnologías han ido evolucionando a lo largo de los años, conllevando nuevas técnicas de codificación y desarrollando así nuevos formatos que permiten una transmisión más optimizada y compatible con los distintos dispositivos de los usuarios finales.

La codificación dentro un sistema puede realizarse de dos formas diferentes: a tiempo real o previamente a su almacenado y posterior visualización. Esta diferencia puede observarse en los diferentes tipos de Streaming:

- ❖ Para Streaming Live es necesaria la codificación del contenido en tiempo real.
- ❖ Para Streaming bajo demanda (VoD) puede codificarse el contenido previamente y almacenarlo en un servidor, por ejemplo, a la espera de que los usuarios finales hagan peticiones para visualizarlo.

Utilizar codificación previa, tiene la ventaja de que los contenidos no poseen restricciones de tiempo real, cosa que permite la implementación de una codificación más eficiente. Sin embargo, presenta un inconveniente bastante importante: la limitación de flexibilidad, que implica que un vídeo codificado no pueda adaptarse a los canales con tasa de bits variable o a los clientes que tienen capacidad de visualización diferente a la originalmente codificada.

El hecho de añadir un módulo de codificación en los sistemas de Streaming, hace que el cliente pueda obtener el contenido deseado en cualquier formato que desee, empleando para ello los codificadores adecuados.

Si a estos codificadores se le suma, además, la característica de la escalabilidad, que permite codificar la señal adaptándola a los recursos de los que disponen los usuarios finales, ya sea a nivel de dispositivo o de velocidad de transmisión de su propia Red, se asegura que el contenido pueda visualizarse con la calidad y la resolución adecuados adaptándose así a los recursos de los que dispone cada uno de los usuarios finales.

3.2. Escalabilidad

La mejora y el avance de las nuevas tecnologías han traído consigo la aparición de nuevas posibilidades en prácticamente todos los ámbitos, incluido en el de codificación.

Una de ellas fue la escalabilidad, que está definida como la habilidad para extraer una escala de representaciones en términos de calidad, resolución o contenido a través del truncamiento del stream codificado completo en ciertos puntos.

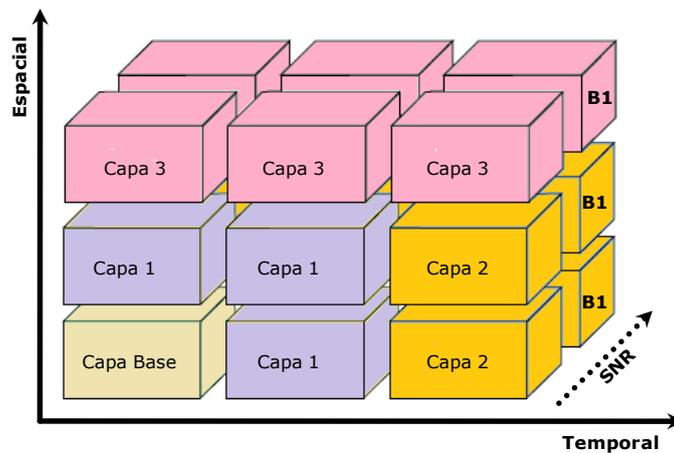


Fig. 3.1 Estructura de un bitstream

Esta necesidad surgió gracias a la posibilidad de visualizar contenidos a través de dispositivos completamente diferentes entre sí, tales como televisores, ordenadores, teléfonos móviles, etc., que requerían la posibilidad de transmitir contenidos en distintas calidades y resoluciones, con el fin de optimizar el proceso de transmisión en el mayor grado posible.

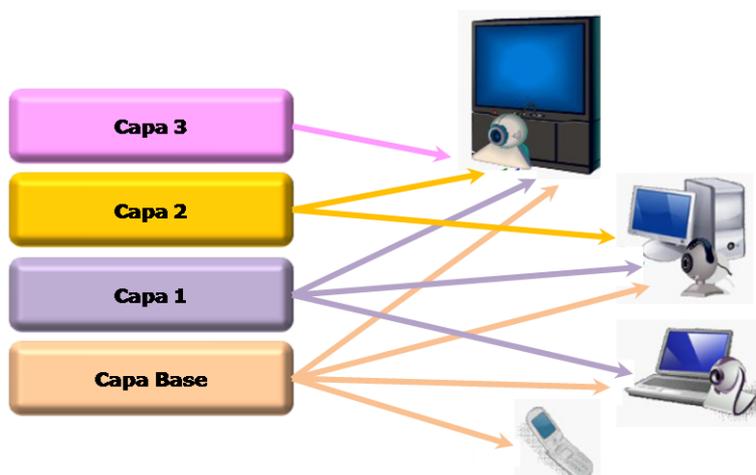


Fig. 3.2 Adaptación de formato

El hecho de que los métodos de codificación basados en el contenido y en la forma de onda busquen optimizar la eficiencia del proceso de codificación para una tasa de bits fija es un problema cuando se quiere acceder a un mismo contenido a través de dispositivos diferentes.

Esto es debido a que un vídeo codificado a una tasa de bits determinada puede ser bajado en tiempo real para su visualización en un terminal conectado a una red con una velocidad de transmisión igual o superior a la del vídeo.

En cambio, si la velocidad de transmisión de la red es menor que la tasa de bits a la que está codificado el vídeo, no será posible visualizarlo en tiempo real ya que el terminal no podrá recibir suficientes bits a tiempo como para mostrar la secuencia en tiempo real.

Este problema desapareció con la aparición de los codificadores escalables, que permiten ajustar la calidad del vídeo a la velocidad de transmisión de la red para poder mostrar la secuencia en tiempo real y la resolución de éste al de la pantalla del dispositivo desde el cual se va a visualizar el contenido.

Otra de las ventajas que ofrecen es la capacidad de adaptación para niveles de error variable en el canal y la capacidad de procesamiento desigual en los receptores.

Existe, sin embargo, otra alternativa a la codificación escalable, conocida como "simulcast" (difusión simultánea), que consiste en codificar un mismo vídeo tantas veces como calidades y resoluciones diferentes queramos obtener.

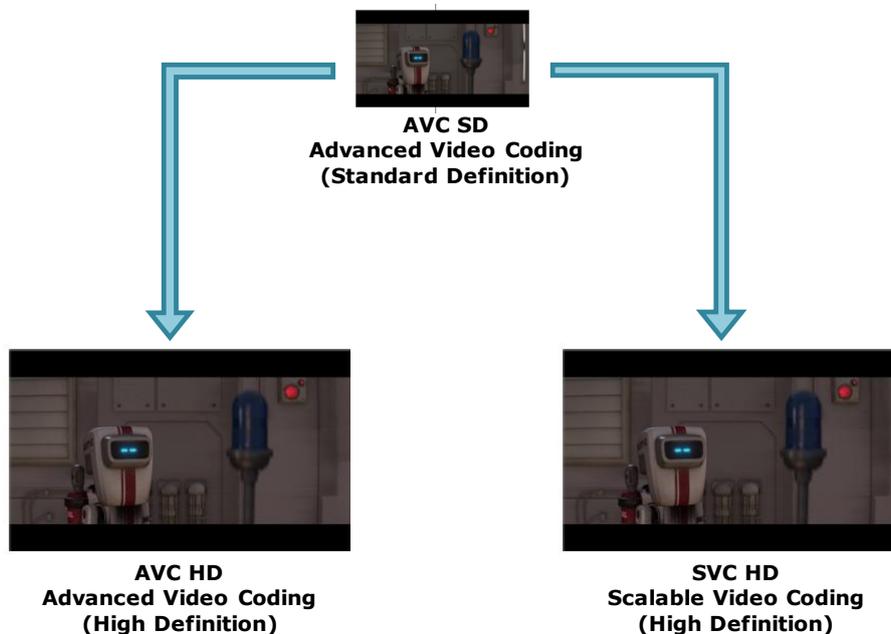


Fig. 3.3 Codificación de Vídeo Escalable (SVC) y Simulcast H.264/AVC

Aunque es una opción que destaca por su simplicidad, es muy ineficiente ya que, dado un flujo de datos con mayor calidad y resolución repetirá esencialmente la misma información que existe en un flujo de menor calidad, junto con información adicional.

Por otra parte, para proveer funcionalidad de escalado, el codificador deberá sacrificar una cierta cantidad de eficiencia en la codificación en comparación con los codificadores no escalables.

Normalmente, la codificación escalable se consigue suministrando versiones múltiples de un vídeo en términos de su resolución de amplitud (escalonamiento en calidad o escalonamiento SNR), resolución espacial (escalabilidad espacial), resolución temporal (escalabilidad temporal) o una combinación de estos tipos.

3.2.1. Escalabilidad de calidad

También conocida como “escalonamiento SNR”, se define como la representación de una secuencia de vídeo con diferente precisión en los parámetros de color.

En este tipo de escalabilidad, se tienen secuencias de bits con N capas de escalabilidad de calidad. Al decodificar la primera capa (también llamada capa base), se crea una versión de baja calidad de la imagen reconstruida y, según se van decodificando las capas siguientes (llamadas capas de mejora), se va incrementando progresivamente la calidad de imagen reconstruida hasta conseguir la calidad original.

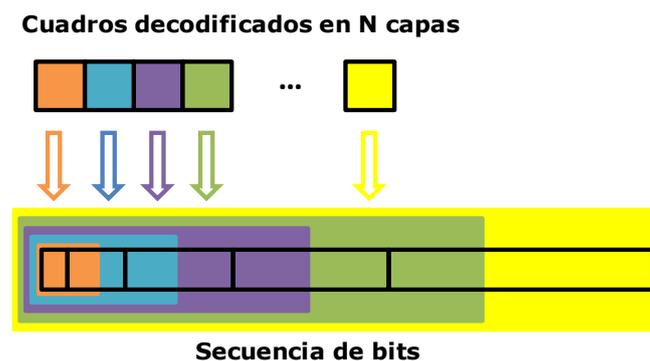


Fig. 3.4 Escalabilidad de calidad

3.2.2. Escalabilidad espacial

Se define como la representación del mismo vídeo en diferentes resoluciones espaciales o tamaños. Decodificando la primera capa, el usuario puede mostrar

una versión preliminar de la imagen decodificada en una resolución inferior y, al decodificar las capas siguientes, se incrementa la resolución espacial hasta llegar a la resolución completa de la imagen original.

En este tipo de escalabilidad, primero debe realizarse una descomposición de la imagen original en múltiples resoluciones donde la imagen de resolución menor es codificada directamente para producir la primera capa (capa base).

Para producir la segunda capa, la imagen decodificada desde la primera capa es interpolada a la segunda resolución y la diferencia entre ésta y la imagen original es codificada. Este proceso se repite para las N capas.

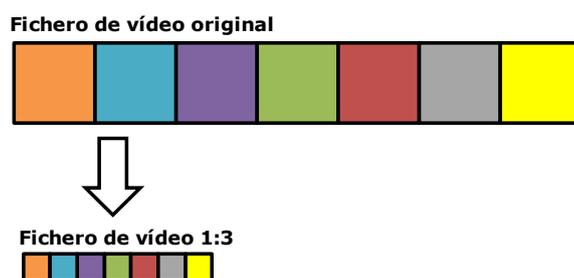


Fig. 3.5 Escalabilidad espacial

3.2.3. Escalabilidad temporal

Se define como la representación del mismo vídeo en diferentes resoluciones. Generalmente, el vídeo escalado temporalmente se codifica utilizando imágenes sobre-muestreadas temporalmente desde una capa inferior como predicción para una capa superior.

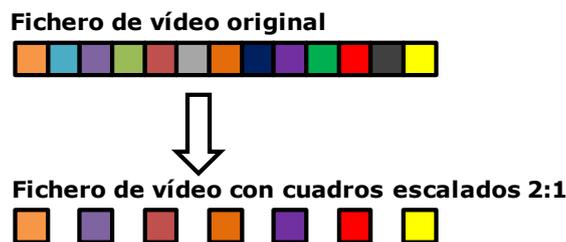


Fig. 3.6 Escalabilidad temporal

3.3. Parámetros de codificación

Para la correcta visualización y adaptación al medio del flujo de vídeo emitido y de los contenidos audiovisuales que se servirán bajo demanda, es necesario hacer un estudio de los parámetros fundamentales de codificación.

En caso contrario, podría causar la pérdida de eficacia del sistema provocada, por ejemplo, por servir contenidos con un factor de compresión muy pequeño, causando que el buffer tarde demasiado en llenarse y, por tanto, la transmisión mediante Streaming sea completamente ineficiente.

En función del ámbito, se puede diferenciar entre dos tipos: parámetros de vídeo y parámetros de audio.

3.3.1. Parámetros de vídeo

Los parámetros de vídeo a tener en cuenta serán los siguientes:

- ❖ Tasa de fotogramas o framerate: es el número de imágenes (de la sucesión de imágenes que forman el vídeo) que se visualizan por segundo. Para que el ojo humano pueda visualizar movimiento, este parámetro debe ser mayor a 20 fps.
- ❖ Resolución: es el tamaño del vídeo, normalmente expresado en píxeles. La resolución del video codificado se calcula a partir de la resolución inicial del video especificada (anchura x altura).
- ❖ Profundidad del color: cantidad de bits necesarios para codificar el color de un píxel. Por ejemplo, si se utilizan 8 bits para codificar los colores de cada píxel, éste podría tener 256 colores.
- ❖ Bitrate o tasa de bits: define el número de bits que se transmiten por unidad de tiempo en un sistema de comunicación digital.
- ❖ Ancho de banda o velocidad de transmisión de la red: es la cantidad de información que se puede enviar a la red en un momento dado.

La relación entre el ancho de banda y la tasa de bits es fundamental ya que el primer parámetro limita al segundo: una tasa de bits mayor que el ancho de banda supondría un colapso en la codificación ya que no se tendría el espacio físico para esa velocidad.

3.3.2. Parámetros de audio

Los parámetros de audio a tener en cuenta serán los siguientes:

- ❖ Número de canales: número de señales de audio simultáneas que contiene un flujo de datos. Puede ser tener un canal (mono), dos canales (estéreo) o varios (multicanal).

- ❖ Frecuencia de muestreo: este parámetro determina la velocidad a la que se toman muestras de la señal analógica. Este parámetro determina el ancho de banda de la señal, ya que éste tiene que ser como máximo la mitad de la frecuencia de muestreo, para que pueda ser recuperada de forma correcta a partir de sus muestras.

Aunque para aplicaciones con una alta calidad de audio o profesionales se utilizan frecuencias de muestreo de 48 o 96 kHz, generalmente suele ser suficiente frecuencias de muestreo más bajas (entre 40 y 44 kHz).

- ❖ Tasa de bits: es el número de bits por segundo que salen del codificador en el proceso de codificación. Una tasa binaria muy elevada implica que el número de bits a almacenar o a transmitir sea demasiado elevado para el canal.

CAPÍTULO 4. COMPRESIÓN

4.1. Introducción

Se entiende por compresión el proceso de conversión de un sistema de datos de origen a otro sistema de datos de destino, siguiendo unas reglas previamente definidas y con el fin de transmitir la misma información mediante medios diferentes.

La realización de transmisión mediante Streaming aporta muchas ventajas a los usuarios finales, especialmente en lo que a disminución de tiempos de espera para visualizar contenidos se refiere. Así pues, la compresión en Streaming, nace de la necesidad de de minimizar dichos tiempos tanto como fuera posible.

Las distintas tecnologías han ido evolucionando a lo largo de los años, conllevando nuevas técnicas de compresión, reduciendo significativamente la información de los archivos a transmitir y garantizando, a su vez, un buen nivel de calidad.

Debido a las características de la transmisión mediante tecnología Streaming, un menor tamaño del contenido implica que se descargue más rápido y, por lo tanto, que tenga un mayor número de posibilidades para conseguir una transmisión de datos en tiempo real para velocidades de transmisión de la red más bajas y/o irregulares.

La razón principal de incluir este módulo en el diseño de la arquitectura del sistema, es el de poder comprimir los datos para poder lograr una mayor eficiencia, ya que el vídeo en bruto requiere un ancho de banda considerable.

4.2. Tipos de compresión

El primer aspecto que se debe tener en cuenta para escoger el formato tanto de audio como de vídeo, es el tipo de compresión de cada uno. Existen dos tipos de compresión: sin pérdida y con pérdida.

4.2.1. Compresión sin pérdida

Este tipo de codificadores conservan los datos originales y aseguran que las imágenes sean las mismas después de la compresión y posterior

descompresión. En estos sistemas se intenta que el codificador extraiga la redundancia de la señal y envíe solo la entropía⁸ al decodificador.

Estos codificadores utilizan normalmente el sistema RLE⁹, consistente en descartar regiones o bloques de similares colores entre imagen e imagen. Esta técnica funciona bien con imágenes generadas por ordenador, donde existen áreas de color uniforme.

Sin embargo, las técnicas de compresión sin pérdida, no son, en general, muy efectivas con el vídeo digital, ya que éste tiene pocas áreas de color continuo y está formado por numerosas variaciones de color.

Este tipo de compresión se caracteriza por tener unos ratios de compresión mucho más bajos (típicamente de 1:2) pero un nivel de calidad mucho más elevado.

4.2.2. Compresión con pérdida

Los codificadores con este tipo de compresión intentan eliminar información de las imágenes de forma que sea lo más inapreciable posible para los usuarios finales. Se intenta eliminar la información irrelevante o no tan crítica para el observador antes de analizar los componentes importantes de la señal.

Solo la entropía es almacenada o transmitida, de forma que el decodificador calcula la redundancia con la señal recibida. La cantidad de información perdida depende del grado de compresión y es proporcional a la disminución de calidad.

Para la transmisión por Streaming, es necesario utilizar codificadores con pérdida ya que, aunque los codificadores sin pérdida garantizan una mejor calidad, el tamaño de los archivos que generan es demasiado elevado.

4.3. Compresión de audio

Además de en la eliminación de información innecesaria, las tecnologías de compresión del audio se basan en el aprovechamiento de las limitaciones del sistema auditivo humano.

Es decir, una vez se ha eliminado toda la información innecesaria para la transmisión mediante diferentes técnicas, pasa a eliminarse aquella cuya ausencia no es capaz de percibir el oído humano. De esta forma pueden llegarse a conseguir ratios de compresión mucho más altos sin afectar, en ningún caso, a la calidad del contenido.

⁸ Cantidad de información (medida en bits) necesaria para codificar un dato.

⁹ Run Length Encoding

En el entorno profesional, los estándares y formatos más extendidos son aquellos incluidos en la familia MPEG y los definidos por Dolby, que utilizan métodos de codificación y compresión basadas en el umbral de audición y el enmascaramiento.

El umbral de audición se define como la intensidad mínima de sonido capaz de percibir el oído humano, el cual no es capaz de percibir sonidos por debajo de un nivel determinado que depende de la frecuencia.

El enmascaramiento es el proceso por el cual el umbral de audición para un sonido aumenta por la presencia de otro sonido. Existen dos tipos: frecuencial y temporal.

- ❖ Enmascaramiento frecuencial: es la disminución de la sonoridad de un tono a cierta frecuencia en presencia de otro tono simultáneo a una frecuencia diferente.

Cuando el oído humano es expuesto a varios sonidos con frecuencias diferentes y de forma simultánea, existe la posibilidad de que uno de ellos impida que los demás puedan ser percibidos.

- ❖ Enmascaramiento temporal: se produce cuando dos sonidos llegan al oído humano muy próximos en el tiempo entre sí. Esto provoca que el tono más intenso enmascare al más débil de forma que este último resulte inaudible.

4.3.1. Compresores de audio

Actualmente existe un amplio abanico de Software de codificación y compresión, como por ejemplo, VLC Media Player, que permiten una gran variedad de códecs de audio, tales como MP3, WAV¹⁰, WMA¹¹, FLAC¹², etc.

Debido al objetivo del sistema del que es objeto el presente documento y a los motivos explicados en el apartado anterior, se descartan los formatos que tienen una peor relación calidad-compresión, requieren la instalación de códecs específicos en la mayoría de reproductores y/o generan archivos demasiado grandes impidiendo así, la optimización de la transmisión por Streaming.

Así pues, se plantean dos opciones: MP3 y AAC¹³.

¹⁰ Wave Audio Format.

¹¹ Windows Media Audio.

¹² Free Lossless Audio Codec.

¹³ Advanced Audio Coding.

4.3.1.1. MP3

El formato MP3 identifica archivos que contienen sonido en formato MPEG Audio Stream, Layer III. Se trata de un formato no propietario que tiene la capacidad de almacenar audio con calidad similar a la de un disco compacto.

Actualmente es el formato más utilizado y extendido para la transmisión de audio mediante Streaming, debido a la reducción del tamaño de los archivos generados (normalmente con una tasa de 1:10) y su gran relación calidad-compresión.

Aunque en los últimos años ha sido parcialmente desbancado por el formato AAC, dado que éste último genera archivos de un tamaño considerablemente menor a MP3, manteniendo una calidad de sonido mayor a una misma tasa de bits y una calidad de sonido similar a la mitad de tasa de bits, MP3 sigue manteniéndose como estándar.

4.3.1.2. AAC

AAC es un formato de audio digital con un algoritmo de compresión con pérdidas, que corresponde al estándar internacional "ISO/IEC 13818-7" como una extensión de MPEG-2.

Aunque no es el estándar de codificación de audio para Streaming, se encuentra en el núcleo del MPEG-4, 3GPP y 3GPP2 y es el códec de audio de elección para Internet, conexiones inalámbricas y de radiodifusión digital, debido a su alto rendimiento y nivel de calidad.

Entre sus múltiples características, cabe destacar las siguientes:

- ❖ Utiliza codificación frecuencial de bits variable (VBR¹⁴) que modifica el número de bits utilizados por segundo para codificar datos de audio en función de la complejidad de la transmisión de audio en un momento determinado.
- ❖ Permite sonidos polifónicos con un máximo de 48 canales y ofrece frecuencias de muestreo que varían de 8 Hz a 96 kHz, en contraposición con las frecuencias de MP3 que varían de 16 a 48 kHz.
- ❖ Utiliza un algoritmo de codificación de banda ancha de audio que produce archivos pequeños con una mejor calidad y requiere menos recursos del sistema para codificar y decodificar.
- ❖ Se basa en la eliminación de redundancias de la señal acústica, así como en la compresión mediante la transformada del coseno discreta modificada (MDCT).

¹⁴ Variable Bitrate.

4.4. Compresión de vídeo

Las señales de vídeo son una sucesión de imágenes estáticas, reproducidas a una velocidad lo suficientemente rápida como para que el ojo humano las vea como una imagen continua en movimiento y no detecte transiciones ni espacios entre ellas.

Esta elevada velocidad implica que se transmitan una gran cantidad de imágenes por segundo, con lo que las variaciones entre dos imágenes consecutivas son prácticamente inexistentes.

La detección de toda esta información redundante es aprovechada en el proceso de compresión ya que no se transmite, consiguiendo de esta forma reducir aún más el volumen de información a enviar.

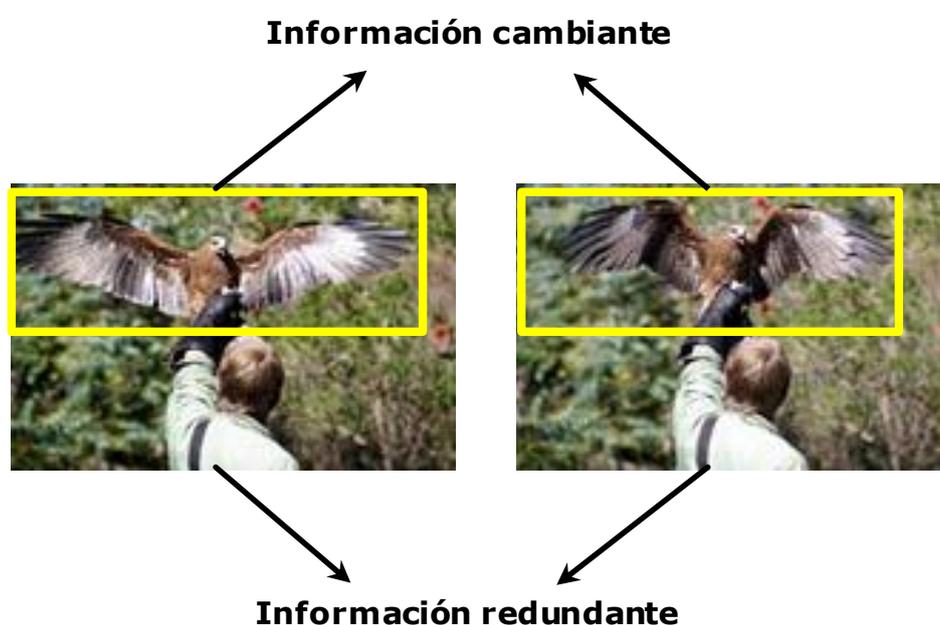


Fig. 4.1 Información redundante y cambiante entre imágenes consecutivas

Generalmente y, debido a los motivos comentados en este apartado, también existe mucha información redundante entre grupos de imágenes en las que existe una mayor separación entre ellas.



Fig. 4.2 Información redundante y cambiante entre grupos de imágenes consecutivas

Con el objetivo de eliminar la información redundante que existe en las secuencias de imágenes que forman los vídeos, se utilizan principalmente las siguientes técnicas: redundancia espacial, redundancia estadística y redundancia temporal.

❖ Redundancia espacial

Aprovechando que los píxeles adyacentes tienen información y características similares, se minimiza la duplicación de los datos en cada imagen. Esto se realiza mediante la Transformada Discreta del Coseno (DCT).

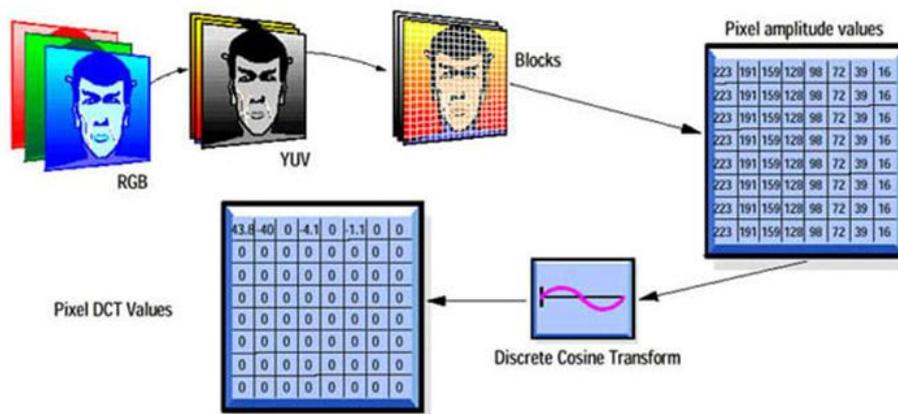


Fig. 4.3 Redundancia espacial

❖ Redundancia estadística

La información más común se codifica en menos bits debido a la aparición de ciertos patrones especiales determinados que aparecen con mayor frecuencia que otros.

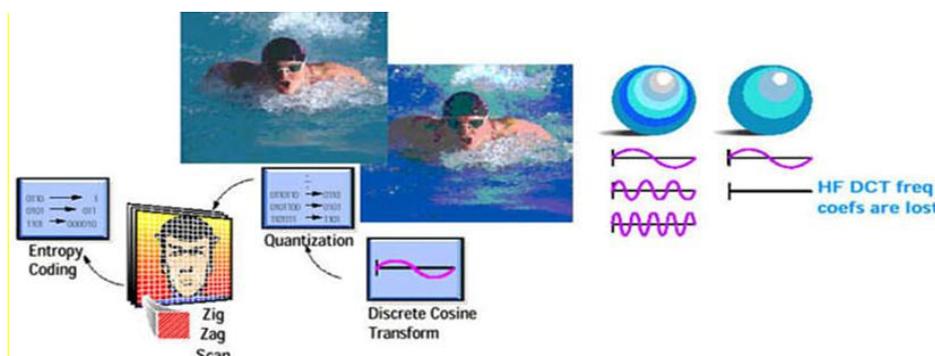


Fig. 4.4 Redundancia estadística

❖ Redundancia temporal

En este tipo de redundancia se utiliza la técnica de Compensación del Movimiento para identificar el desplazamiento de un macrobloque¹⁵ determinado en el cuadro actual respecto a la posición que tenía en el cuadro de referencia, aprovechando que las imágenes consecutivas tienen información parecida y relacionada entre sí.

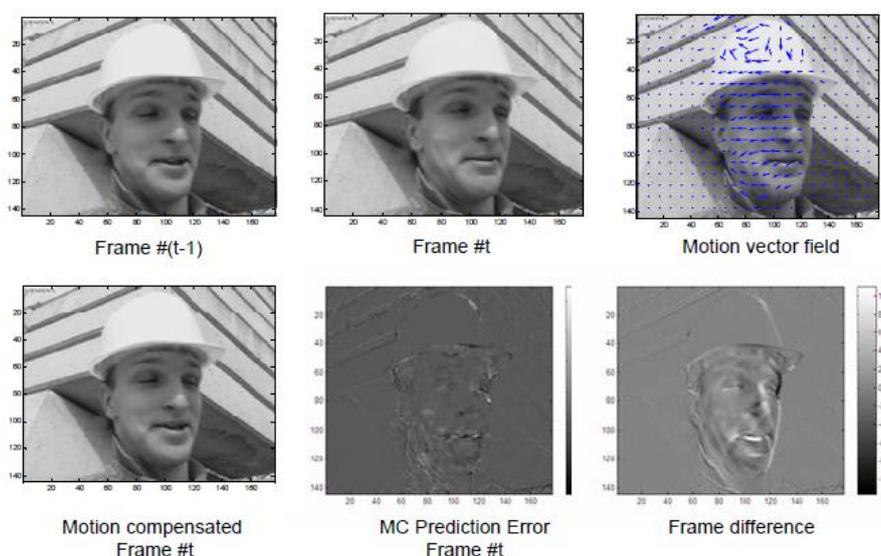


Fig. 4.5 Técnica de Compensación de movimiento

¹⁵ Unidad de compensación de movimiento. Contiene una sección del componente de luminancia y los componentes de crominancia espacial correspondientes.

Una vez detectado ese desplazamiento, se buscan los macrobloques del cuadro a codificar en el cuadro de referencia y se codifican las diferencias entre imágenes.

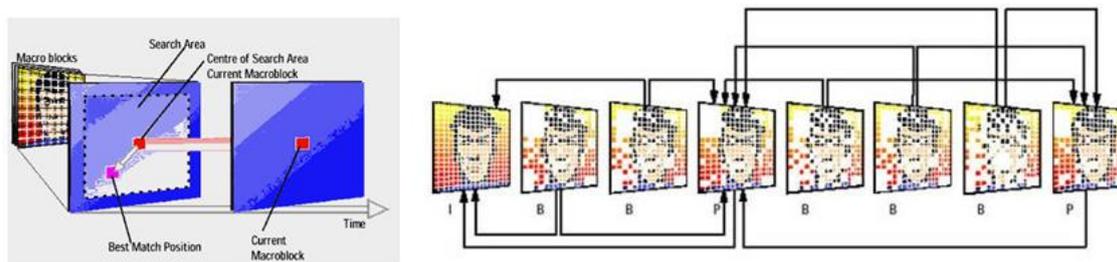


Fig. 4.6 Redundancia temporal

4.4.1. Compresores de vídeo

De la misma forma que para audio, existe también un amplio abanico de Software de codificación y compresión, que permiten una gran variedad de códecs de audio, tales como Theora, MJPEG, WMV¹⁶, etc.

Debido al objetivo del sistema del que es objeto el presente documento y a los motivos explicados en el apartado anterior, se descartan los formatos que tienen una peor relación calidad-compresión, requieren la instalación de códecs específicos en la mayoría de reproductores y/o generan archivos demasiado grandes impidiendo así, la optimización de la transmisión por Streaming.

Así pues, se plantea una única opción: H.264.

4.4.1.1. H.264

H.264 es un códec de alta compresión capaz de proporcionar una buena calidad de imagen con tasas binarias notablemente inferiores a los estándares previos, como H.263, además de no incrementar la complejidad del diseño.

Este estándar contiene una serie de nuevas características respecto a los estándares previos que permiten comprimir el vídeo de forma mucho más eficaz que los antiguos formatos (hasta un 50% más que sus predecesores) y proporciona una mayor flexibilidad para su aplicación en una amplia variedad de entornos de red.

Permite degradar la calidad de forma inteligente para asegurar una reproducción continua del vídeo. Si se dan problemas de congestión de red, el

¹⁶ Windows Media Video

servidor de vídeo disminuye el número de fotogramas que está enviando para mantener la calidad e ir llenando el buffer mínimamente.

Si las condiciones empeoran, el servidor deja de mandar fotogramas de vídeo, pero mantiene la calidad del audio. Finalmente, si la calidad del audio empieza a degradarse, el cliente reconstruye de manera inteligente las secuencias que tiene almacenadas para no perder la calidad.

El estándar H.264 se creó con la finalidad de poder alcanzar una compresión más alta, para compensar las restricciones de capacidad de canal limitada, manteniendo la calidad del vídeo.

Su perfecta adecuación para la transmisión vía tecnología Streaming se basa en las siguientes características básicas:

- ❖ Es una técnica bastante resistente a soportar errores por el empleo de redes poco fiables, como Internet.
- ❖ Proporciona una flexibilidad considerable para ser utilizada en aplicaciones en tiempo real con un retardo bajo, una mayor resolución en la estimación de movimiento¹⁷ que sus antecesores y una selección de imagen de múltiple referencia.
- ❖ Proporciona una compresión sin pérdidas apreciables en la calidad del vídeo alcanzando una reducción de hasta un 50% de la tasa de bits respecto a los estándares predecesores y un 80% del formato original, permitiendo así visualizar vídeos con mucha más calidad que con los protocolos anteriores para un mismo ancho de banda.

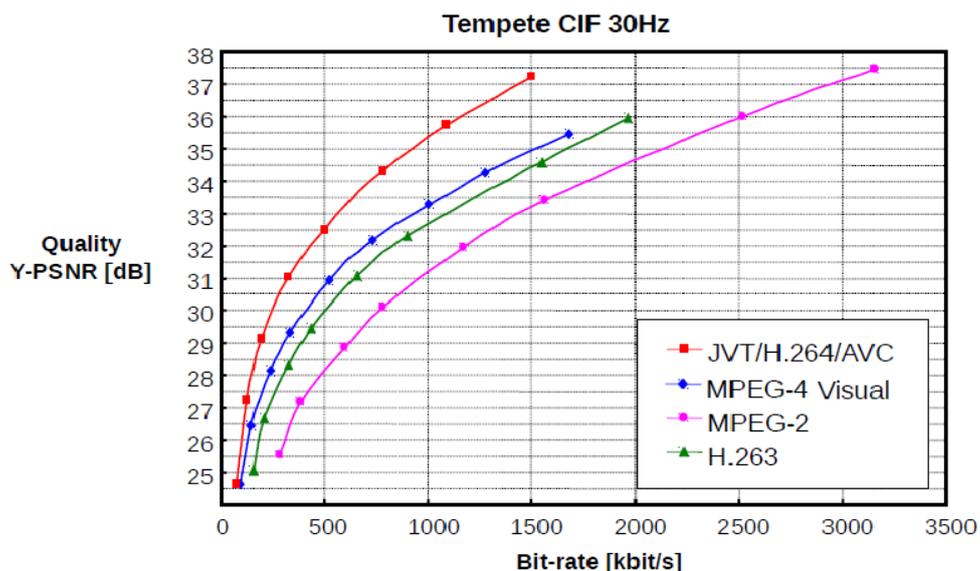


Fig. 4.7 Comparativa de formatos

¹⁷ Identificación de los movimientos entre dos cuadros (vectores de movimiento) de la imagen a codificar respecto a una o más imágenes de referencia.

En las tablas siguientes, puede observarse la mejora de MPEG-4 AVC / H.264 respecto a sus predecesores:

Codificador	MPEG-4 ASP	H.263	MPEG-2
H.264/AVC	38,62%	48,8%	64,46%
MPEG-4 ASP	-	16,65%	42,95%
H.263	-	-	30,61%

Tabla 4.1. Comparativa de ratio de mejora

Predecesores	H.264
El canal de transmisión funciona o no, no hay término medio.	Baja degradación cuando se pierden las partes “correctas” del bitstream.
Los canales de transmisión tienen un rendimiento (throughput) fijado.	Adaptación de la tasa de bits al rendimiento (throughput) del canal.
Solo se utiliza un único formato, definido por el estándar.	Adaptación del formato para otras extensiones compatibles.
No existe adaptación de potencia dado que no se contemplan dispositivos portátiles.	Adaptación de potencia para la compensación entre el tiempo de ejecución y calidad.

Tabla 4.2. Comparativa de mejoras respecto a sus predecesores

En este tipo de codificación, la imagen es dividida en bloques. Cada muestra de un bloque es predicho utilizando muestras espacialmente vecinas de los bloques codificados previamente.

El proceso de codificación selecciona cuál y cómo serán usadas las muestras para la Intra predicción¹⁸, detallada más adelante en el presente capítulo, la cual es simultáneamente conducida al codificador y decodificador usando la información obtenida por Intra predicción.

A continuación, puede verse el diagrama de bloques del codificador H.264:

¹⁸ Predicción derivada de las muestras decodificadas de la misma porción decodificada.

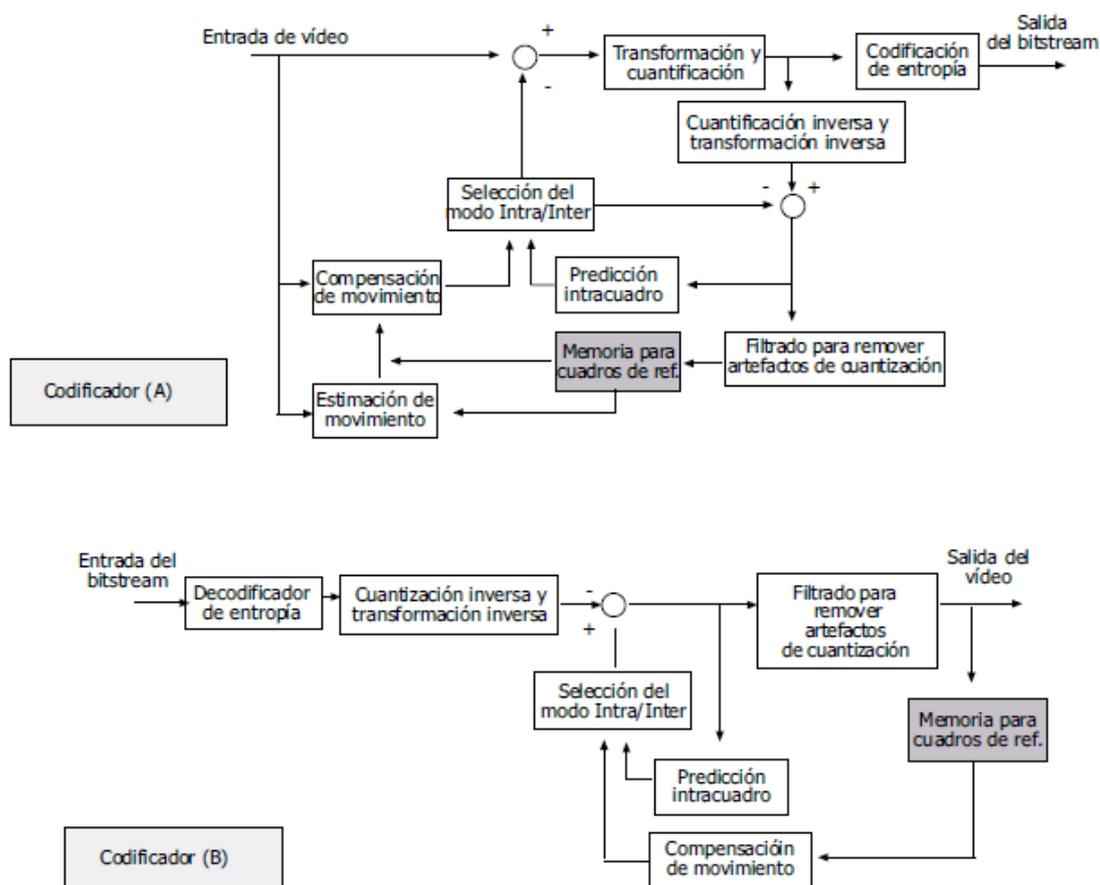


Fig. 4.8 Diagrama de bloques del codificador H.264

El estándar H.264 incluye un codificador de entropía mejorado, una mayor precisión para la compensación de movimiento y métodos más efectivos de protección contra errores, además de dos técnicas de predicción: intracuarto e intercuarto.

❖ Predicción Intra cuadro (Intra)

El estándar H.264 utiliza este tipo de predicción para codificar bloques o macrobloques de referencia y reducir la cantidad de bits codificados. En la predicción Intra cuadro se forma un predictor basado en los píxeles frontera del macrobloque dentro del mismo cuadro y se codifica la señal residual (error) entre el bloque actual y la predicción, disminuyendo de esta forma la cantidad de bits que representan a dicho bloque considerablemente.

❖ Predicción Inter cuadro (Inter)

La predicción Inter cuadro se basa en la utilización de vectores de movimiento recibidos por parte del decodificador para llevar a cabo la misma predicción que realizó el codificador. Es decir, mediante mecanismos de estimación de

movimiento, se realiza una selección de data en movimiento y se le aplica un desplazamiento espacial a todas las muestras del bloque.

Esta predicción puede realizarse de forma más efectiva gracias a la característica innovadora de H.264, que permite asignar diferentes tamaños de bloques en función de la cantidad de movimiento que haya en los distintos frames.

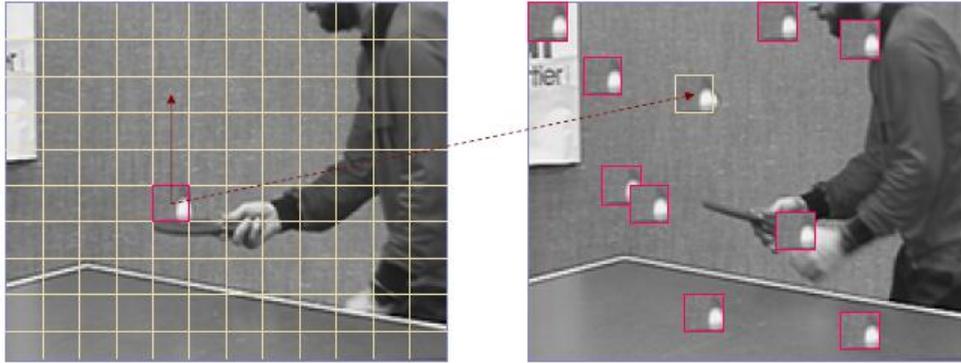


Fig. 4.9 Inter predicción

4.4.1.2. H.265

Ante la creciente demanda del mercado por trabajar con vídeos de altas resoluciones se comenzó, hace aproximadamente dos años, a desarrollar y definir un nuevo estándar denominado temporalmente como H.265 o H.265/HEVC¹⁹.

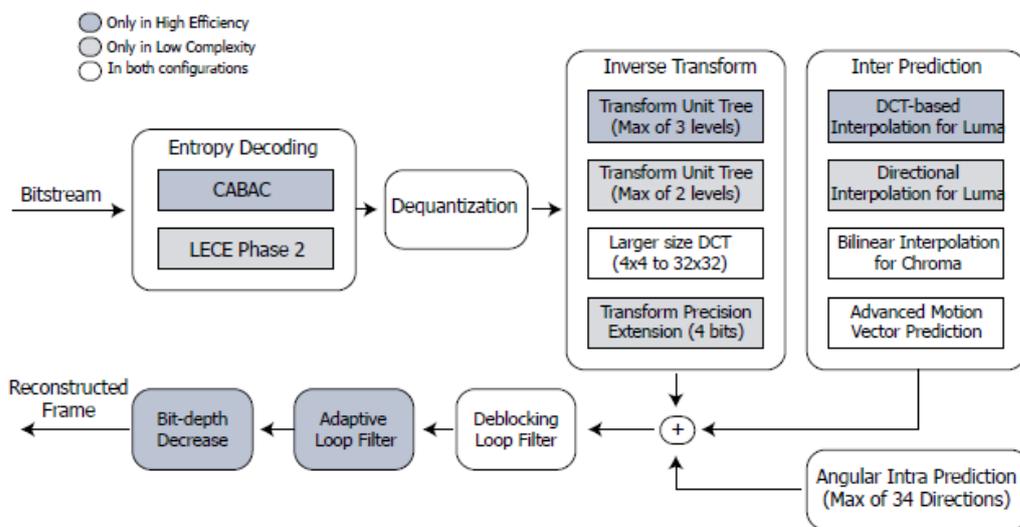


Fig. 4.10 Estructura de codificación de H.265

¹⁹ High Efficiency Video Coding.

Este estándar integrará nuevas herramientas que reducirán el tiempo de procesamiento, permitirán la reducción de la tasa de bits a la mitad para una codificación de alta calidad sin la degradación de imagen y doblarán el ratio de compresión, entre otras ventajas, mejorando así el proceso global de codificación.

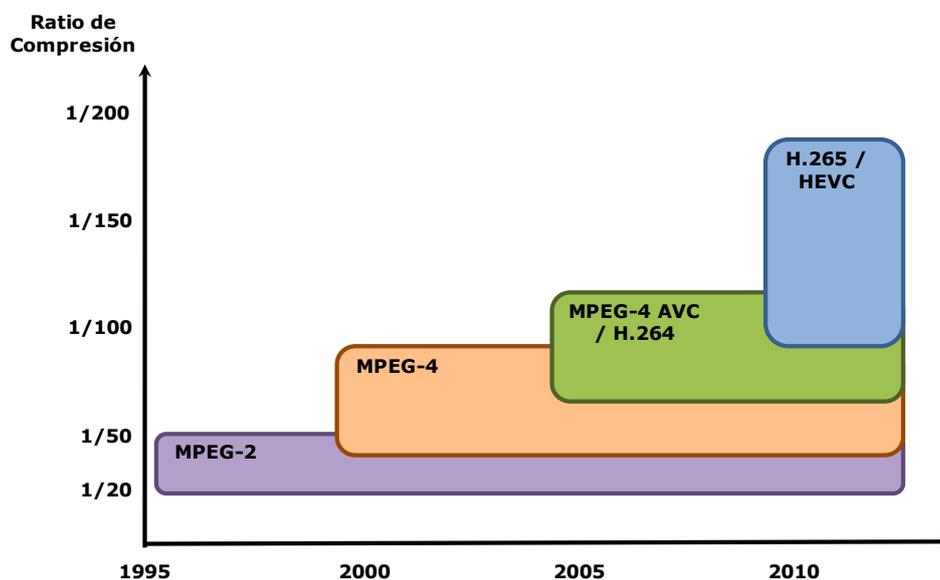


Fig. 4.11 Evolución de las tecnologías de compresión de vídeo

Se espera que este estándar pueda ser estandarizado a lo largo del año 2013 y pueda empezar a implantarse durante los años siguientes en los nuevos dispositivos del sector audiovisual.

CAPÍTULO 5. ENCAPSULAMIENTO Y TRANSPORTE

5.1. Introducción

Se entiende por “encapsulamiento” el proceso mediante el cual se forman paquetes de datos que se puedan administrar y rastrear, incorporando una unidad del protocolo de datos de una capa dentro de una unidad del protocolo de datos de una capa inferior, y con el fin de enviar los datos a través de una red.

De la misma forma que para audio y vídeo, existe también un amplio abanico de Software que permiten una gran variedad de opciones de encapsulamiento tales como AVI²⁰, WebM, MP4, etc.

Debido al objetivo del sistema del que es objeto el presente documento, se descartan los formatos de encapsulamiento que generan archivos con un tamaño demasiado grande, que no soportan transmisión vía Streaming y/o que requieren la instalación de códecs específicos en la mayoría de reproductores o navegadores del mercado.

Así pues, se plantean dos opciones: MP4 y MPEG-TS.

5.1.1. MP4

MP4 es un formato de archivo contenedor que forma parte del estándar MPEG-4. Se utiliza, de forma bastante extendida, para la distribución de audio y vídeo que cumplen con el estándar MPEG-4 (por ejemplo, H.264 AVC para vídeo y AAC para audio), aunque también puede almacenar otro tipo de datos como subtítulos, información de capítulos e imágenes fijas.

La extensión asociada a los archivos que cumplen este estándar es .MP4, aunque también pueden encontrarse archivos de audio que lleven la extensión .M4A, que es la extensión adoptada por Apple para la distribución de música en iTunes y su reproductor iPod.

5.1.2. MPEG-TS

Formato de encapsulamiento para audio, vídeo y datos especificados en los estándares MPEG-2. Los distintos flujos binarios se comprimen

²⁰ Audio Video Interleave.

independientemente formando cada uno de ellos un stream elemental (ES²¹) y, a su vez, se estructuran en paquetes (PES²²).

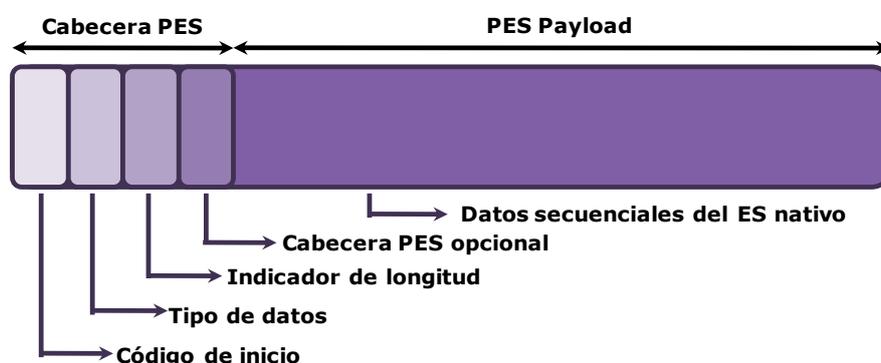


Fig. 5.1 Estructura básica del PES

En el estándar MPEG-2 existen dos métodos de constituir una trama de datos: Program Stream (PS) y Transport Stream (TS).

5.1.2.1. Program Stream

Tipo de trama formada por paquetes, con una cabecera por cada uno de ellos, definida para el almacenamiento de datos en medios con una baja influencia de errores, como por ejemplo DVDs, discos duros, etc.

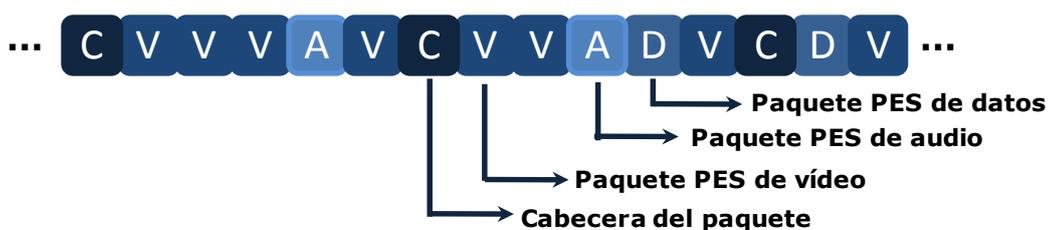


Fig. 5.2 Multiplexación de Program Stream

Las tramas Program Stream son muy vulnerables a errores, ya que están formadas por paquetes de gran longitud, factor que implica que el equipo de recepción no pueda saber dónde termina un paquete y empieza el siguiente.

Si el campo de cabecera de un paquete se pierde o tiene errores, el equipo de recepción perderá el sincronismo de datos con la consiguiente pérdida de paquete, ocasionando pérdidas significativas en la trama de audio y vídeo.

²¹ Elementary Stream

²² Packetised Elementary Stream

5.1.2.2. Transport Stream

Es un tipo de trama definida en MPEG-2 para transmisión en medios con alta influencia de errores, como pueden ser las aplicaciones broadcast, la transmisión de televisión digital e Internet.

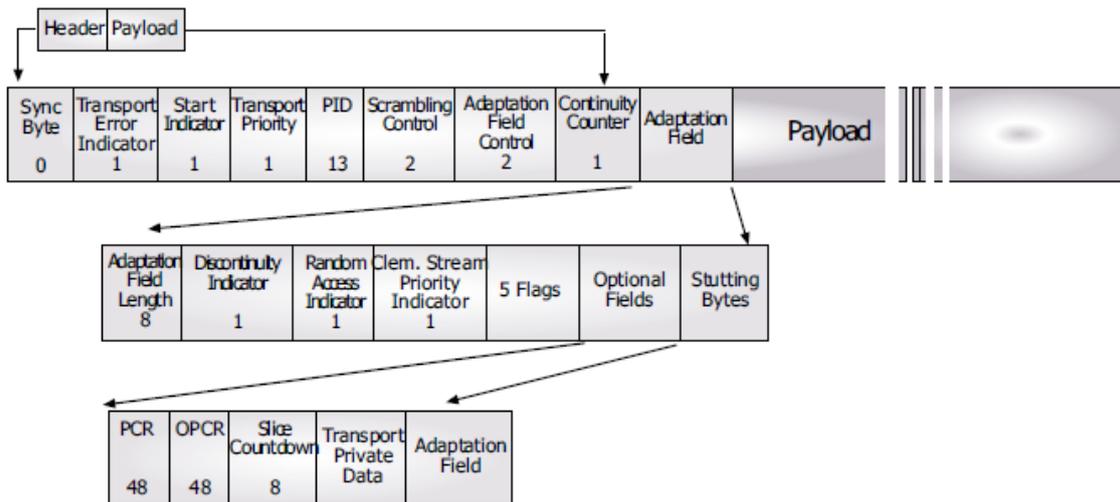


Fig. 5.3 Estructura de un paquete de transporte

- ❖ Sync byte: sincroniza el decodificador con el paquete e indica que comienza un paquete de transporte.
- ❖ Transport error indicator: indica si existen errores en la transmisión.
- ❖ Payload unit indicator: indica el inicio del Payload.
- ❖ PID: permite la identificación unívoca de los paquetes de los ES.
- ❖ Transport scrambling control: indica si hay datos encriptados en el payload.
- ❖ Adaptation field control: indica si en la cabecera existe Adaptation Field.
- ❖ Continuity counter: campo que incrementa su valor entre paquetes consecutivos que pertenecen al mismo ES, que permite al decodificador saber si se ha perdido algún paquete.

La multiplexación en Transport Stream está basada en paquetes de tamaño fijo y tiene la capacidad de corregir los problemas provocados por la existencia de errores, ya que dispone de métodos para su detección.

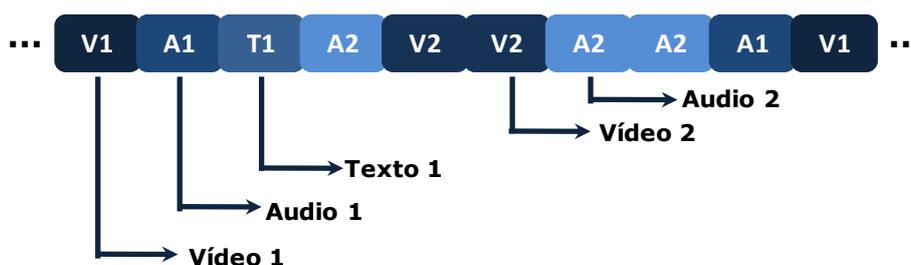


Fig. 5.4 Proceso de multiplexación

5.2. Protocolos de transporte

El protocolo de transporte es el encargado de que se efectúe la transferencia de datos libre de errores entre el emisor y el receptor, aunque no estén directamente conectados, así como de mantener el flujo de la red. Su objetivo es establecer una transmisión fiable entre la máquina origen y la máquina destino, independientemente de la red física que haya en medio.

Debido al objetivo del sistema del que es objeto el presente documento, se descartan aquellos protocolos que requieran la existencia de una conexión entre emisor y receptor o que ocasionen problemas en la transmisión de vídeo en tiempo real, como por ejemplo TCP²³.

Estos protocolos deberán cumplir con los criterios expuestos a continuación para permitir la realización de la transmisión en tiempo real de manera óptima:

- ❖ Debe incluir en su cabecera un número de secuencia, incrementándolo en la unidad para cada paquete y así determinar si hay pérdidas.
- ❖ Debe incluir también una marca temporal (timestamp) que permita saber el momento en el cual se generó la información que contiene el paquete y así medir el retardo de transferencia entre emisor y receptor y el jitter²⁴.
- ❖ El emisor y el receptor deben poder comunicarse entre ellos para compartir información de cómo visualiza cada uno la emisión y así poder la aplicación reaccionar a los estados de la red.
- ❖ Así como en una transferencia HTTP o FTP es inaceptable la pérdida de un paquete, en una transmisión audiovisual y en directo es asumible tener la pérdida de una cantidad pequeña de información, ya que el problema que causaría tener al final todos los paquetes correctos en el receptor sería peor que la propia pérdida de éstos.

Así pues se plantean tres opciones: UDP²⁵, RTP²⁶ y RTSP²⁷.

²³ Transport Control Protocol.

²⁴ Desfase entre un paquete y otro que hace que hayan gaps en la reproducción de la información.

5.2.1. UDP

Es un protocolo de transporte utilizado en Streaming no orientado a conexión, es decir, que no requiere la existencia de una conexión entre emisor y receptor para efectuar la transmisión.

Permite el transporte de información de vídeo y audio a gran velocidad, aunque también se utiliza para el transporte de datos. Dado que no dispone de mecanismos de control de la conexión entre emisor y receptor, el emisor se limita a enviar los datagramas con una dirección y un puerto.

5.2.2. RTP

RTP es un protocolo especialmente diseñado para el transporte de audio y vídeo en tiempo real. Utiliza UDP como mecanismo de transporte de paquetes, añadiendo una cabecera a cada uno de ellos.



Fig. 5.5 Protocolo RTP

Este protocolo soporta transmisiones tanto unicast como multicast, así como diferentes formatos de audio y vídeo estándar y múltiples velocidades binarias, que permiten a los receptores codificar solo determinadas partes del total.

En general RTP añade funcionalidades a UDP y su uso está extendido especialmente para las aplicaciones de transmisión multimedia e IPTV. Se caracteriza por ser un protocolo adaptable, es decir, por adaptar su formato al codificador concreto que transporta.

Sin embargo, existe una cabecera común para todos los codificadores, que contiene los siguientes campos destacables:

²⁵ User Datagram Protocol.

²⁶ Real-Time Transport Protocol.

²⁷ Real-Time Streaming Protocol.

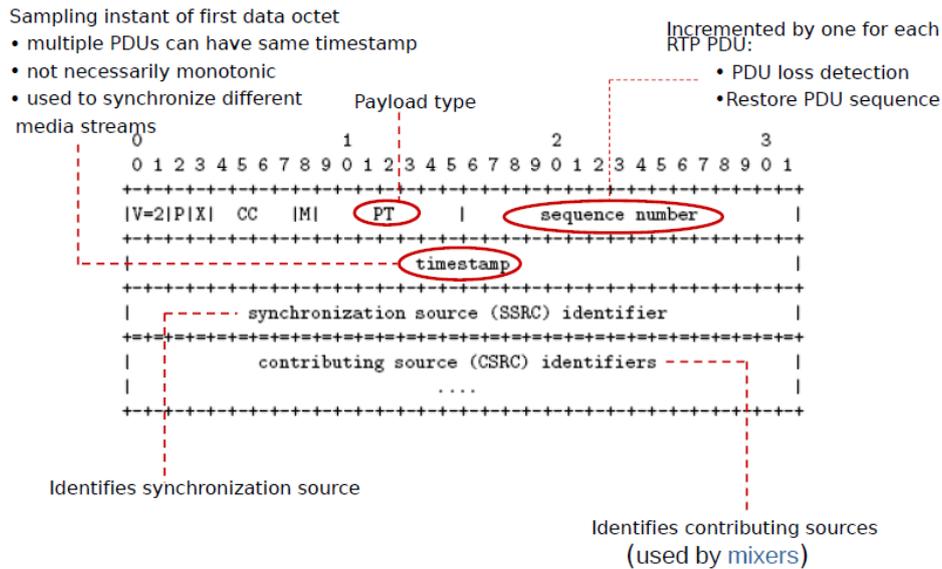


Fig. 5.6 Cabecera RTP

- ❖ Versión: actualmente la 2.
- ❖ Cantidad de contribuyentes: número de fuentes de emisión que llegan al mezclador (mezcla de diferentes señales para transmitir un solo flujo de stream, máximo 15).
- ❖ Payload type: identifica el códec.
- ❖ Número de secuencia: se incrementa por cada paquete que se transmite. Se escoge un número inicial aleatorio para evitar ataques criptográficos en caso de que RTP vaya cifrado.
- ❖ Timestamp: instante de muestreo del primer byte de datos transportado por el paquete. Se suele escoger, igual que el número de secuencia, un número inicial aleatorio.
- ❖ Id contibuyente: identifica el contribuyente al que pertenece ese paquete.

5.2.3. RTSP

El protocolo de Streaming en tiempo real (RTSP) tiene como objetivo entregar contenidos como secuencia unicast. Se creó específicamente para controlar la entrega de datos en tiempo real, como por ejemplo contenido de audio y vídeo.

Se implementa a través de un protocolo de control que funciona conjuntamente con el Protocolo en Tiempo Real (RTP) de entrega de datos para proporcionar contenido a los clientes.

Este tipo de protocolo tiene las características siguientes:

- ❖ Permite cualquier descripción de sesión.
- ❖ Controla pistas de audio y presentaciones de vídeo.
- ❖ Permite la edición digital remota del contenido.

Su funcionamiento se basa en mensajes de texto. Los mensajes principales de solicitud que se pueden mandar pueden ser del tipo:

- ❖ SETUP: el servidor asigna recursos y establece una sesión RTSP.
- ❖ PLAY: empieza la transmisión de datos.
- ❖ PAUSE: detiene temporalmente la transmisión.
- ❖ TEARDOWN: libera recursos y termina la sesión RTSP.

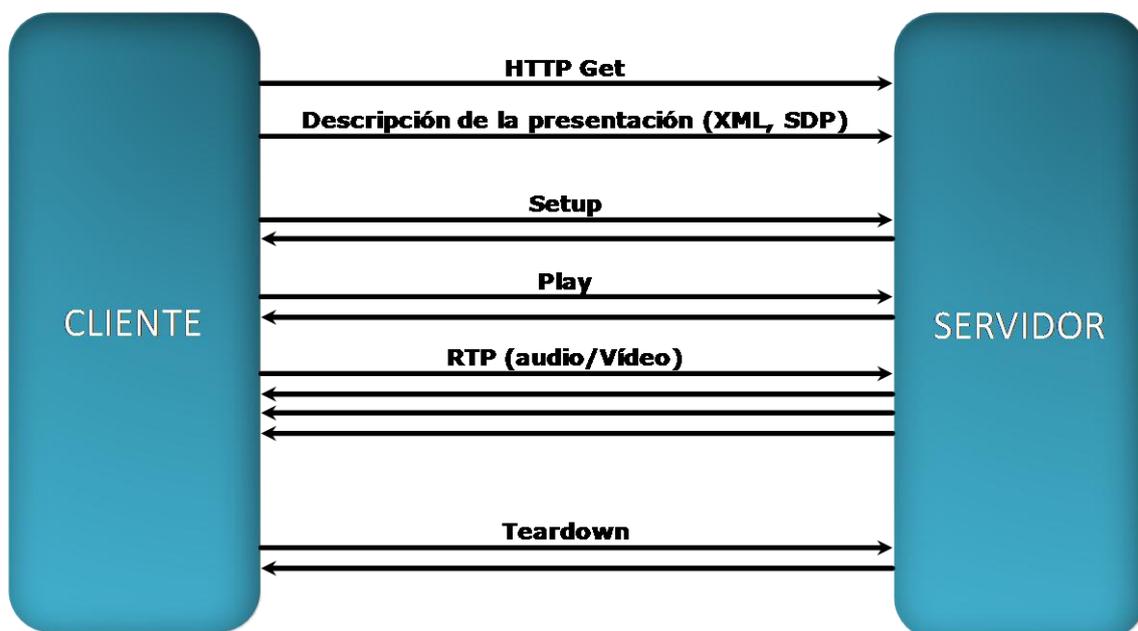


Fig. 5.7 Esquema de funcionamiento de RTSP

SDP (Session Description Protocol) es un protocolo utilizado para describir los parámetros de inicialización de los flujos multimedia, cubriendo aspectos como anuncio de sesión, invitación a sesión y negociación de parámetros.

Este protocolo no se encarga de entregar los contenidos propiamente dichos, sino de entablar una negociación entre las entidades que intervienen en la sesión como tipo de contenido, formato, y todos los demás parámetros asociados, conocidos como “perfil de sesión”.

CAPÍTULO 6. TRANSMISIÓN STREAMING

6.1. Introducción

Tal y como se ha visto a lo largo de los capítulos anteriores, el sistema de transmisión Streaming puede dividirse, de cara al usuario final, en los siguientes módulos:

- ❖ Ingesta: donde el cliente introduce en el sistema los contenidos que desea distribuir a los distintos usuarios finales.
- ❖ Distribución: donde se envía y reparte el contenido ingestado de acuerdo a las peticiones de los usuarios finales.
- ❖ Publicación: donde se muestra (publica) el contenido ingestado de forma que pueda ser accesible para los usuarios finales.
- ❖ Visualización: donde los usuarios finales reproducen y visualizan el contenido deseado.

6.2. Soluciones de mercado

Existen varias maneras diferentes de realizar este tipo de transmisión ya que, actualmente, es la manera de visualizar contenido multimedia más extendida y utilizada gracias a que elimina la necesidad de descargar un contenido por completo para poder visualizarlo, minimizando así el tiempo de espera y la ocupación en disco.

Como se desea realizar una transmisión Streaming tanto en directo como Bajo Demanda, es recomendable utilizar un software que ofrezca la posibilidad de realizar ambas y no tener que utilizar un software para cada tipo de Streaming ya que, de esta manera, el flujo de información provendrá de un único software simplificando así el sistema y reduciendo los requisitos Software.

Considerando las diferentes soluciones posibles que ofrece el mercado hoy en día, se ha optado por utilizar el software VLC Media Player²⁸ ya que además de ser un software libre y gratuito, permite más opciones de configuración (calidad, resolución, formato...) que el resto de soluciones freeware del mercado tales como, por ejemplo, Ustream²⁹ y además la posibilidad de simular una fuente de emisión a diferencia de FFMpeg³⁰.

²⁸ <http://www.videolan.org/vlc/>

²⁹ <http://www.ustream.tv/>

³⁰ <http://ffmpeg.org/>

Aunque el software FFMpeg permite ajustar varios parámetros de codificación y compresión con el fin de optimizar la transmisión, se ha descartado la opción de utilizarlo ya que VLC Media Player ofrece además de eso una interfaz gráfica, que facilita las configuraciones a realizar ofreciendo así la opción de no tener que ejecutar obligatoriamente el programa en modo consola, y la posibilidad de envío de señal a través de él, simulando así la ingesta de señal de un canal de televisión.

En el caso de que este sistema se comercializara, debería considerarse un cambio de Software a uno de pago, bien a una de las soluciones que se ofrecen actualmente o bien a una solución hecha a medida.

La razón de este cambio es que se debe poder garantizar al cliente unos niveles de servicio, mantenimiento, desarrollo y soporte mínimos, que no pueden asegurarse en versiones libres.

6.3. VLC Media Player

VLC Media Player es un software libre y gratuito compatible con todos los Sistemas Operativos del mercado (Windows, Linux, etc.), diseñado para la reproducción de archivos multimedia como DVD, Audio CD, webcams y varios protocolos de Streaming sin necesidad de descargar e instalar códecs.

Este Software ofrece múltiples opciones de configuración en lo que a compresión y codificación se refiere, permitiendo de esta forma modificar parámetros como la calidad, la velocidad, el encapsulamiento, la tasa de bits, el formato, etc. tanto para el audio como para el vídeo y tanto desde el emisor como desde el receptor, asegurando así que el usuario final pueda visualizar el contenido de acuerdo a unos parámetros definidos que él mismo puede modificar.

VLC Media Player permite la utilización de una amplia variedad de códecs tanto de audio como de vídeo, así como diversos formatos contenedores³¹ donde almacenar dicha información de la forma más adecuada posible en función de los formatos de audio y vídeo seleccionados para la transmisión.

Otra de las características de VLC es que, además de la interfaz gráfica, ofrece la posibilidad de programar estos parámetros mediante código y evitar así recorrer varios submenús para conseguir la configuración deseada.

³¹ Formato de archivo que almacena información tanto de audio como de vídeo (y si se deseara, de subtítulos), con el fin de no tener información de audio y vídeo por separado.

6.3.1. Configuración de emisión

Para poder emitir una señal de vídeo, se debe descargar e instalar la última versión del software VLC Media Player de la página oficial y realizar la configuración necesaria en el ordenador que hará de emisor, siguiendo los pasos que se describen a continuación:

- a. Seleccionar el menú Medio de VLC Media Player y la opción de Abrir dispositivo de captura.



Fig. 6.1 Paso 1: Abrir dispositivo de captura

- b. Una vez seleccionada esta opción, se abrirá la siguiente ventana, donde deberá seleccionarse el nombre de dispositivo de vídeo y de audio con los que se va a capturar el contenido a emitir. Tanto los parámetros de la cámara (brillo, contraste, zoom...) como los del micrófono (agudos, volumen...) pueden modificarse haciendo clic en los botones de Configurar que aparecen en este menú.

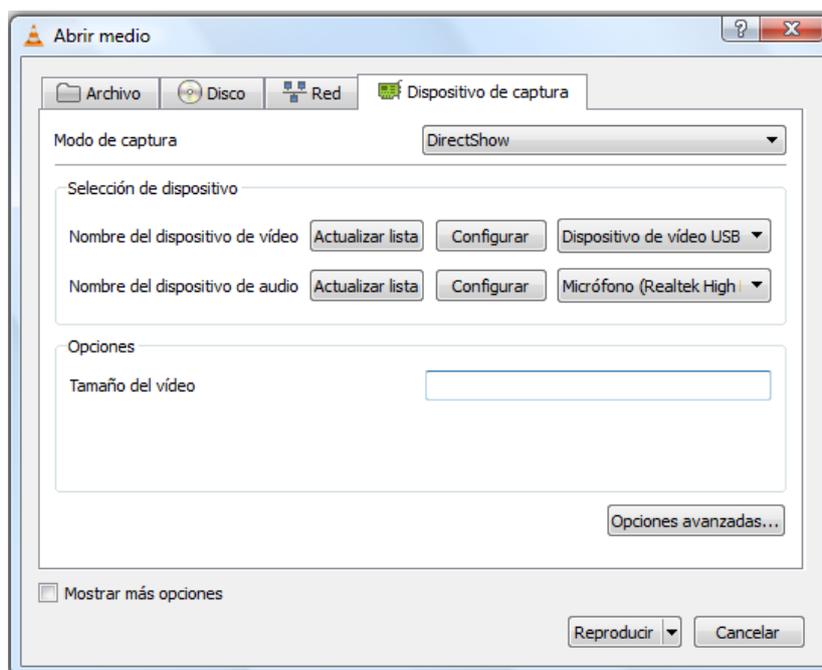


Fig. 6.2 Paso 2: Configuración de dispositivos de audio y vídeo

Desde aquí también pueden configurarse otros parámetros tales como la duración del vídeo, el momento de inicio (horas-minutos-segundos), la proporción de la imagen, la tasa de fotogramas de entrada, etc.

- c. Una vez se ha terminado de configurar los parámetros de este menú de la forma que el emisor considere adecuada, debe seleccionarse la opción de Emitir del desplegable.

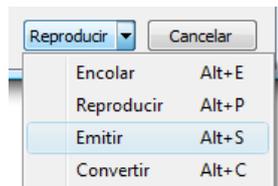


Fig. 6.3 Paso 3: Seleccionar “Emitir”

- d. Al seleccionar la opción de Emitir se abrirá una nueva ventana donde se podrá seleccionar el protocolo de transmisión, en este caso y por los motivos ya mencionados, se ha optado por utilizar RTSP. Una vez seleccionado el protocolo, debe hacerse clic en el botón de Añadir.

Al seleccionar dicho botón, aparecerá una nueva pestaña donde el usuario deberá escribir la dirección IP a la que quiere transmitir la señal.

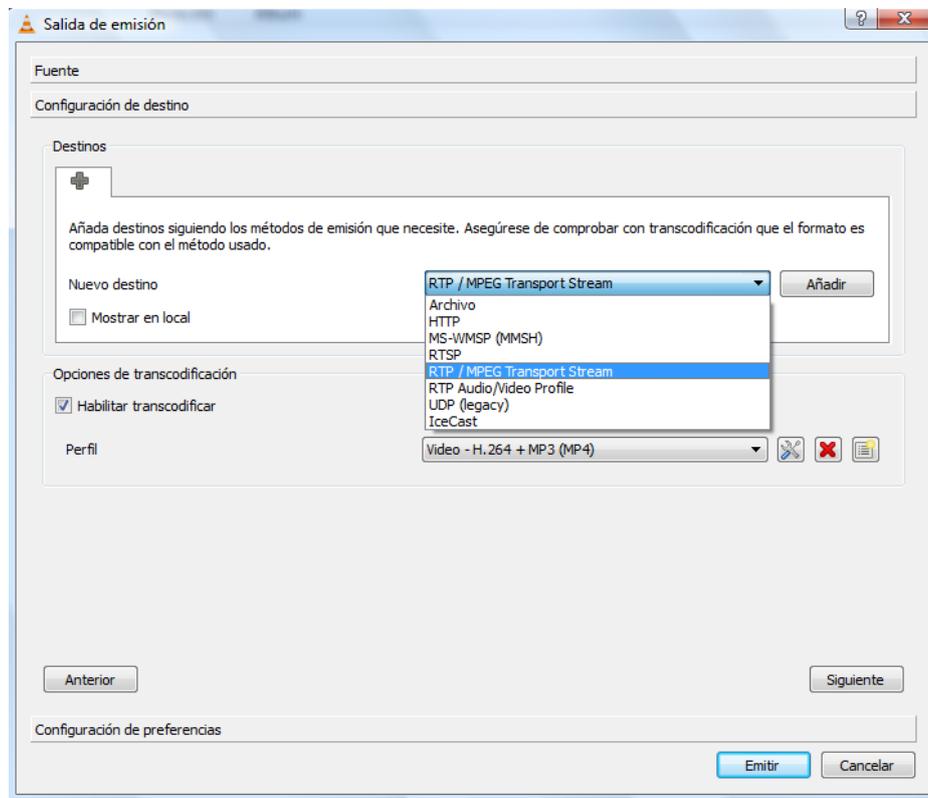


Fig. 6.4 Paso 4: Selección del protocolo de transmisión

Este menú proporciona también la opción de poder ver la señal que se está emitiendo desde el emisor, habilitando el checkbox de Mostrar en local.

- e. Aquí pueden escogerse los formatos y parámetros a los que se va a codificar la señal de entrada. VLC proporciona unos cuantos perfiles ya definidos que el usuario puede elegir si lo desea o, incluso, editarlos. También puede crear un nuevo perfil que se ajuste a su retransmisión si lo cree necesario.

En el caso de que no quiera codificarse el vídeo, puede deshabilitarse el checkbox Habilitar transcodificar o bien crear un perfil en RAW que nos permita transmitir los datos en crudo.

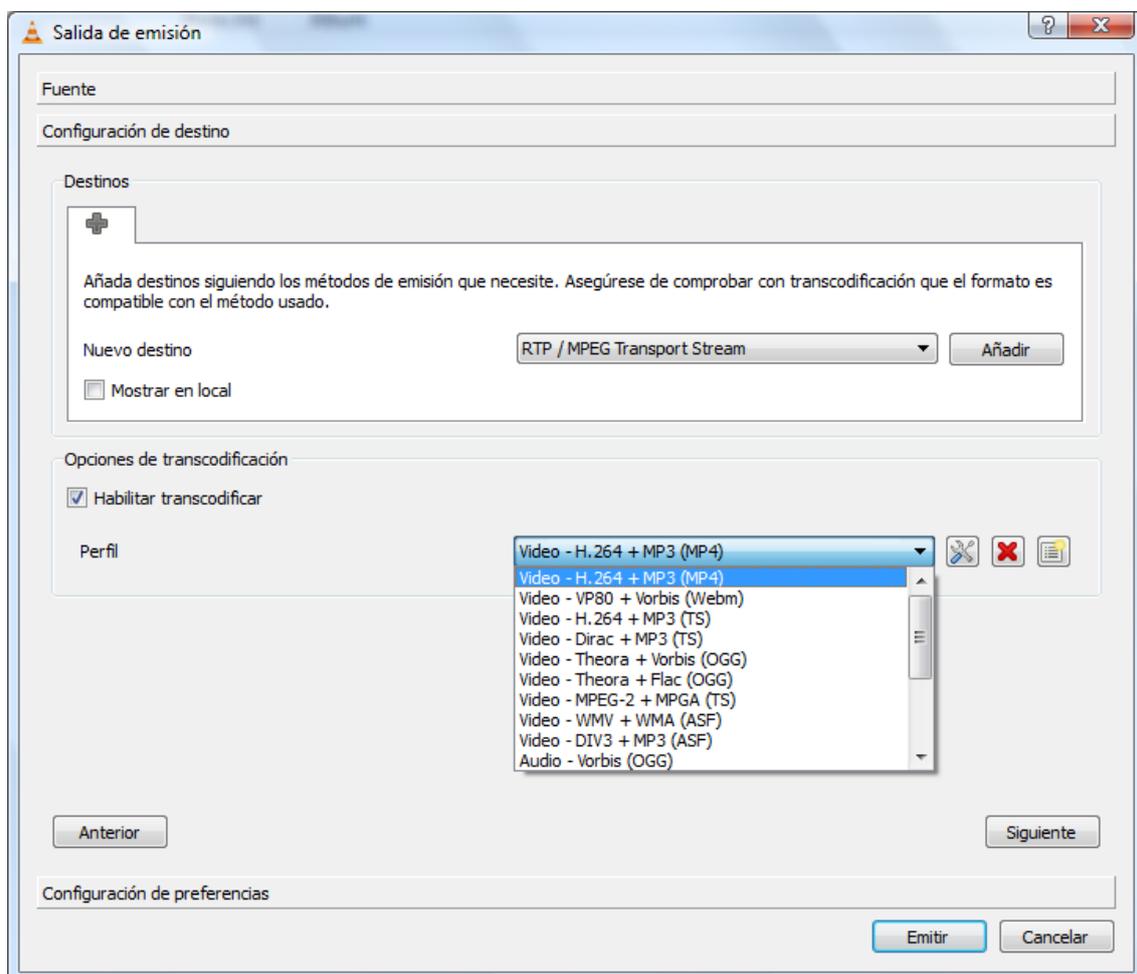


Fig. 6.5 Paso 5: Selección del perfil de codificación

6.3.2. Configuración de recepción

Para poder recibir la señal del emisor, deben seguirse los pasos que se describen a continuación:

1. Seleccionar el menú Medio de VLC Media Player y la opción de Abrir volcado de red.

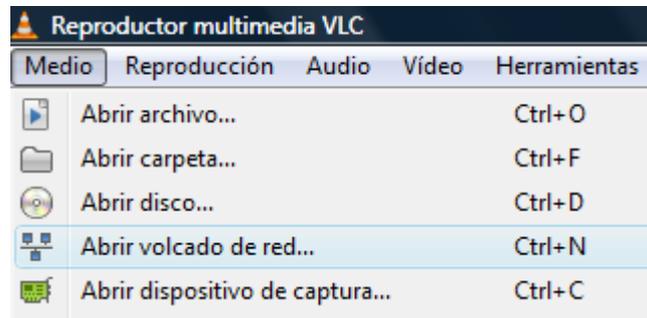


Fig. 6.6 Paso 6: Abrir volcado de red

2. El usuario receptor deberá escribir su propia IP junto con el puerto y el protocolo por el cual está retransmitiendo el emisor y seleccionar el botón de Reproducir. En este caso sería:

```
rtsp://@IP_RECEPTOR:1234
```

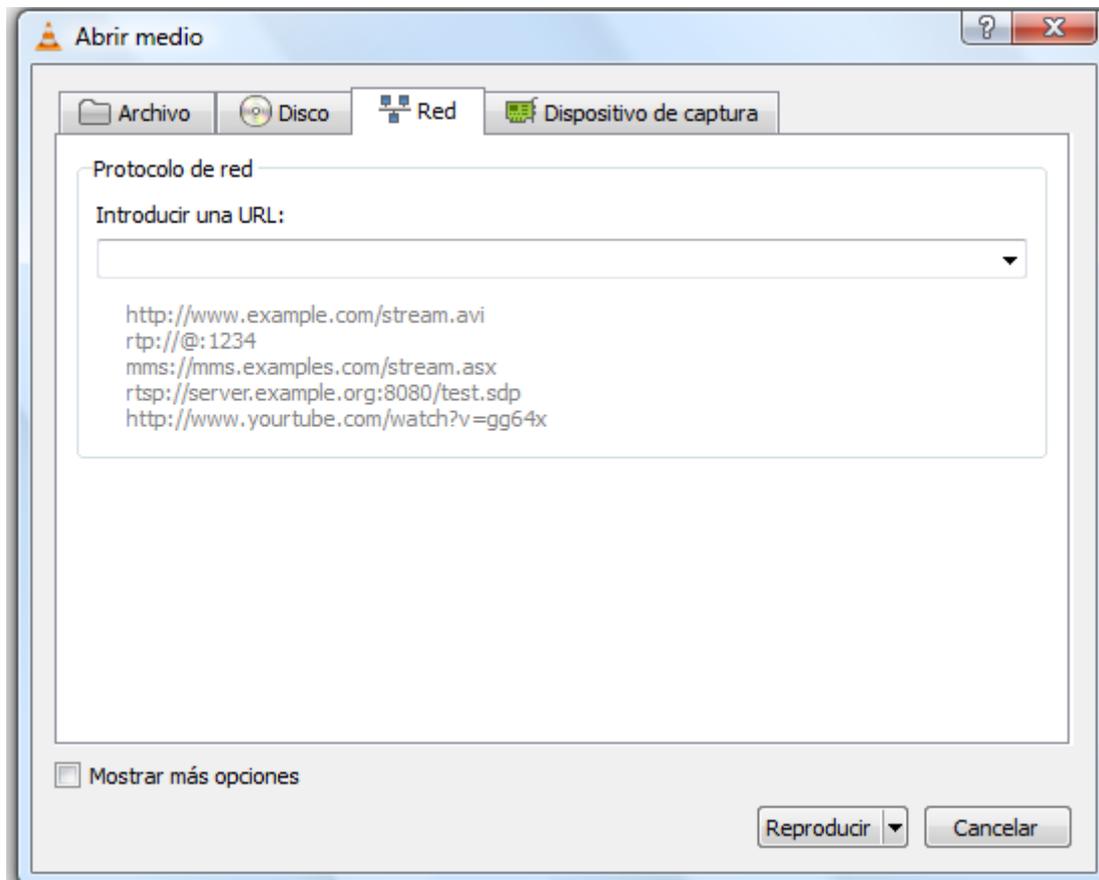


Fig. 6.7 Paso 7: Introducción de la URL

CAPÍTULO 7. SERVIDOR

7.1. Introducción

Se conoce como “servidor” el equipo físico y centro de la infraestructura de la red, formado por dispositivos de red que brindan un servicio a otros dispositivos, conocidos como “clientes”.

Si bien es cierto que cualquier ordenador puede ser un servidor sin necesidad de contar con un Hardware o Software específicos, existen sistemas operativos especializados, como por ejemplo Linux, que han sido específicamente diseñados para optimizar los recursos que se comparten a la red.

Debido a las necesidades, tanto del propio tipo de transmisión como del sistema del que es objeto este proyecto, se requiere el uso de un servidor, capaz de soportar transmisión Streaming, para almacenar, introducir y gestionar los contenidos que el cliente quiera ofrecer a los usuarios.

En principio, para los casos de transmisión Streaming de contenido Bajo Demanda (VoD), no sería necesario contar con un servidor especial para colocar los contenidos multimedia que serán posteriormente descargados desde una web, ya que cualquier servidor normal puede enviar la información de forma que sea el cliente quien se encargue de procesarla para poder mostrarla posteriormente a medida que la va recibiendo.

Sin embargo, el uso de los servidores de Streaming ofrece importantes prestaciones que resultan de gran utilidad al sistema, como por ejemplo, almacenar el contenido, gestionar las conexiones individuales provenientes de los distintos usuarios y mandar archivos de diferentes calidades en función de la velocidad de la Red de la que disponga el usuario que ha realizado la petición, de forma que pueda visualizar el contenido de manera adecuada, sin cortes ni tiempos de espera excesivos.

Para los casos de transmisión Streaming de contenido en directo, sin embargo, es imprescindible utilizar un servidor que soporte este tipo de transmisión para enviarle la señal que el cliente quiere transmitir y que este, a su vez, la envíe a los distintos usuarios a medida que la va recibiendo, posibilitando así el visionado a tiempo real del contenido.

7.2. Características

La elección de un servidor de Streaming debe estar determinada por una serie de criterios técnicos orientados a satisfacer unos requerimientos determinados

en función del tipo de audiencia al que se quiera retransmitir y al tipo de contenido que se vaya a distribuir.

En el caso de este proyecto, estos criterios estarán orientados a un ambiente académico en el sistema de pruebas y a la distribución de contenido multimedia orientada tanto a través de PC como de dispositivos móviles Android. Por tanto, deberemos tener en cuenta los siguientes aspectos:

- ❖ Soporte unicast: todos los servidores de Streaming tienen este tipo de soporte, pero lo que diferencia a uno de otro es la cantidad de conexiones simultáneas que pueda manejar.
- ❖ Soporte multicast: es uno de los aspectos más importantes en la transmisión de Streaming en redes privadas, donde el número de usuarios y la utilización del ancho de banda de la Red son tan prioritarios.
- ❖ Codificación de vídeo MPEG-4: el estándar H.264 se ha establecido como una de las técnicas más convenientes en la generación de Streaming, dada su gran relación compresión – calidad.
- ❖ Modularidad en la arquitectura: se requiere un servidor con una arquitectura modular en la cual sea fácil agregar módulos propios al servidor y brindarle así características extras.
- ❖ Open source: se recomienda la utilización de servidores con código abierto y una comunidad de desarrolladores amplia, de forma que la solución resulte lo más económica posible y exista la posibilidad de recibir cierto soporte.

7.3. Servidor web

Además del servidor mencionado en este capítulo, necesario para el almacenamiento y gestión del contenido multimedia, el sistema objeto del presente documento precisa también un servidor web, necesario para la publicación del contenido a los diferentes usuarios de la Red.

Así pues, un servidor web es un Software que se ejecuta de forma continuada en un ordenador y que tiene como objetivo alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP.

Básicamente, este servidor consta de un intérprete HTTP que se mantiene a la espera de peticiones de usuarios y les responde con el contenido adecuado según sea solicitado. El dispositivo del usuario final, una vez ha recibido el código, lo interpreta y lo muestra en pantalla.

Además, los servidores pueden disponer de un intérprete de otros lenguajes de programación (ASP, PHP, Perl, etc.), que ejecutan código embebido dentro del código HTML de las páginas que contiene el sitio antes de enviar el resultado al usuario.

Para este proyecto, se ha decidido utilizar un servidor web Apache debido a los motivos que se exponen a lo largo del apartado siguiente.

7.3.1. Servicio APACHE

Apache es el servidor web más utilizado por excelencia debido a, entre otros factores, su gran estabilidad, eficiencia, configurabilidad, sencillez y universalidad, siendo capaz de correr en múltiples plataformas y Sistemas Operativos (Windows, Linux, etc.) y de trabajar con una gran cantidad de lenguajes de programación (Perl, PHP, Java, etc.).

Se caracteriza por ser una tecnología completamente gratuita de código abierto y por tener unos ficheros de configuración en un formato simple (ASCII), que permite su modificación mediante editores de texto, y completamente transferibles, facilitando así la clonación del servidor.

Su diseño completamente modular, le permite adaptarse fácilmente tanto a sistemas pequeños y básicos, mediante instalaciones sencillas, como a posibles sistemas comerciales, mediante la incorporación de nuevos módulos, extendiendo así sus funcionalidades y calidad de sus servicios.

Esta característica permite construir un servidor a medida, adaptándose así a las necesidades del cliente y minimizando el consumo de memoria RAM y espacio en el Disco Duro, ya que no deberá mantener módulos que le aporten funcionalidades que no se requieran.

En cuanto a la administración del servidor, puede realizarse mediante línea de comandos, lo cual posibilita su administración remota y ofrece la posibilidad de configurarlo para que ejecute scripts determinados cuando ocurra un error concreto, de forma que se minimice así la necesidad de una administración manual.

7.3.2.1. Configuración APACHE

Para este sistema se ha mantenido la configuración estándar de la distribución APACHE, así como el punto de publicación del servicio, ubicado en la ruta siguiente:

`/var/www/`

En este directorio se alojará todo el contenido que publicará el servidor web de forma que pueda ser visualizado por los usuarios finales, así como las diversas plantillas, imágenes y páginas índice utilizadas para su realización y posterior prestación del servicio.

7.3.2. Servicio FTP

El servicio FTP (File Transfer Protocol) es uno de los más antiguos dentro de Internet, conocido por ser utilizado para transferir información, almacenada en ficheros, de una máquina remota a otra local o viceversa.

Actualmente, existen tres tipos diferentes de FTP:

- ❖ FTP anónimo: cualquier usuario puede descargarse archivos sin necesidad de identificarse previamente.
- ❖ FTP autenticado: los usuarios deben conectarse al servidor mediante un nombre y contraseña válidos para enviar/recibir archivos.
- ❖ FTP embebido: es una opción dentro del FTP anónimo. Se realiza desde las páginas web a través de navegadores.

Normalmente, como regla general, se suelen utilizar el servicio FTP anónimo y el embebido para la descarga de contenidos y, el autenticado, para la subida (transferencia) de contenidos al servidor.

Mediante este servicio, se puede controlar el acceso de los usuarios dentro del sistema de carpetas del servidor gracias a la definición administración de los usuarios y a la asignación de permisos de los diferentes archivos y directorios.

El servicio FTP admite dos modos diferentes de conexión del cliente: activo y pasivo. En modo activo, cuando el cliente FTP inicia una transferencia de datos, el servidor abre una conexión desde el puerto 20 en el servidor para la dirección IP y un puerto aleatorio sin privilegios (mayor que 1024), especificado por el cliente. Esto implica que la máquina cliente debe poder aceptar conexiones de cualquier puerto superior a 1024.

Sin embargo, el hecho de tener configurado un Firewall interno, podría provocar que se rechazaran estas conexiones aleatorias, así que se ha decidido configurar el servicio FTP en modo pasivo con la distribución VSFTPD para mejorar la seguridad del servicio.

Al configurarlo en modo pasivo, el cliente FTP indica que quiere acceder a los datos en modo pasivo y el servidor proporciona la dirección IP y un puerto concreto dentro de un rango definido en la configuración, sin privilegios en el servidor, para que el cliente pueda conectarse y descargar la información requerida.

7.3.2.1. Configuración FTP

Para este sistema, se ha mantenido la configuración estándar alojada en el fichero “vsftpd.conf”, aplicando las modificaciones siguientes:

- ❖ Tal y como se ha comentado en el apartado 5.3.1.1 del presente documento, se ha habilitado una carpeta para este servicio ubicada en:

```
/home/ftp
```

- ❖ Se ha configurado a 5 minutos el tiempo que una sesión puede permanecer abierta sin interacción alguna del usuario, evitando de esta forma que se mantengan hilos inactivos.

```
idle_session_timeout=300
```

- ❖ Se ha configurado el servicio FTP para que se levante de forma automática ante cualquier reinicio del servidor.

Aunque para este sistema no se ha habilitado, debido a que simula el sistema de un radiodifusor único, se recomienda la habilitación del enjaulado de usuarios FTP para sistemas comerciales o de más de un broadcaster.

De modo predeterminado por el servicio VSFTPD, los usuarios del sistema que se autentiquen tienen acceso a otros directorios del sistema fuera de su directorio personal.

Así pues, con el objetivo de que cada usuario solo tenga acceso a su propio directorio personal y lo que este contiene cada vez que se autentifique en el servidor FTP, se deberán establecer “jaulas” para los usuarios.

De esta manera se recluye a los usuarios a solo poder utilizar su propio directorio personal y así evitar el acceso por parte de éstos a los directorios de otros.

```
chroot_local_user=YES  
chroot_list_enable=YES  
chroot_list_file=/etc/vsftpd/chroot_list
```

Con la especificación “*chroot_local_user=YES*”, se establece que el archivo “*chroot_list*” sea la lista de usuarios que quedan excluidos de la función `chroot()`.

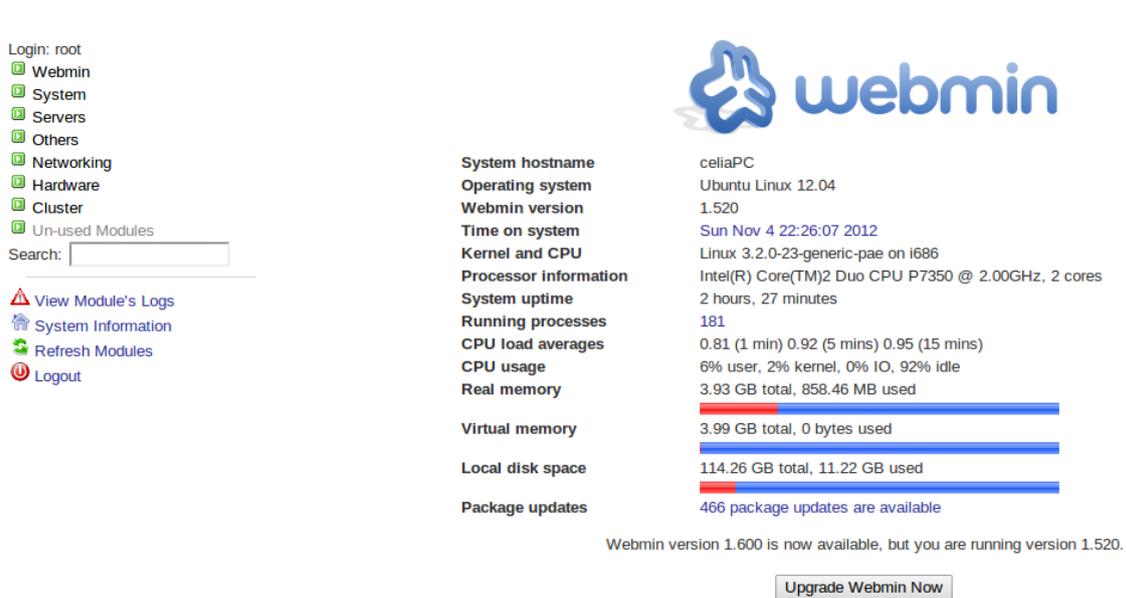
7.4. Webmin

Webmin es una interfaz web para la administración de sistemas de Linux (Unix). Usando cualquier navegador, el usuario administrador puede configurar

las cuentas de usuario, APACHE, DNS, apagar el equipo, compartir archivos, dar permisos para usuarios o grupos, etc.

Además, elimina la necesidad de editar manualmente los ficheros de configuración y permite a los usuarios administradores manejar el sistema desde el propio equipo o remotamente.

Webmin está escrito en Perl y ejecuta tanto su propio proceso como su servidor web, por lo que no es necesario tener instalado APACHE o cualquier otro servidor web, pero convive sin problemas con ellos. Por defecto, se comunica a través del puerto TCP 10.000.



The screenshot shows the Webmin interface. On the left, there is a navigation menu with the following items: Login: root, Webmin, System, Servers, Others, Networking, Hardware, Cluster, Un-used Modules, and a search box. Below the menu are links for View Module's Logs, System Information, Refresh Modules, and Logout. The main content area displays system information for 'cellaPC' running Ubuntu Linux 12.04. The information includes: System hostname (cellaPC), Operating system (Ubuntu Linux 12.04), Webmin version (1.520), Time on system (Sun Nov 4 22:26:07 2012), Kernel and CPU (Linux 3.2.0-23-generic-pae on i686), Processor information (Intel(R) Core(TM)2 Duo CPU P7350 @ 2.00GHz, 2 cores), System uptime (2 hours, 27 minutes), Running processes (181), CPU load averages (0.81 (1 min) 0.92 (5 mins) 0.95 (15 mins)), CPU usage (6% user, 2% kernel, 0% IO, 92% idle), Real memory (3.93 GB total, 858.46 MB used), Virtual memory (3.99 GB total, 0 bytes used), Local disk space (114.26 GB total, 11.22 GB used), and Package updates (466 package updates are available). A notification at the bottom states 'Webmin version 1.600 is now available, but you are running version 1.520.' with an 'Upgrade Webmin Now' button.

Fig. 7.1 Webmin

7.5. Firewall

Para que puedan ser efectivas las configuraciones y funcionalidades detalladas a lo largo de este capítulo, se deben realizar modificaciones en el Firewall del servidor, añadiendo las siguientes reglas:

1. Se habilita la conexión SSH por el puerto 22, para permitir el acceso a la máquina por consola de forma remota.
2. Se habilita la conexión por el puerto 10000 para permitir el acceso a la herramienta gráfica Webmin, que tiene como objetivo poder realizar una administración del servidor de una forma más fácil y amigable a través de su interfaz gráfica.

3. Se habilita la conexión por los puertos 20:21 para el acceso por parte del cliente, la publicación de contenidos y la gestión de las aplicaciones mediante el servicio FTP.
4. Se habilita la conexión HTTP por el puerto 80, para permitir el acceso a Internet.

CAPÍTULO 8. HTML5

8.1. Introducción

Con el objetivo de poder visualizar los flujos de vídeo emitidos y los contenidos audiovisuales almacenados en el servidor, es necesaria la creación de una página web que incluya un reproductor vinculado a dichos flujos y contenidos.

VLC permite incluir mediante su plugin un reproductor compatible con el estándar HTML, considerado el estándar de referencia por excelencia en lo que a programación web se refiere.

Con el fin de garantizar una sincronización lo más eficaz y óptima posible, se ha utilizado VLC como reproductor, aunque HTML permite la introducción de otros reproductores diferentes embebidos.

8.2. Características

El estándar HTML (HyperText Markup Language) hace referencia al lenguaje de marcado³² predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de etiquetas, rodeada por corchetes angulares (<>) y puede incluir un script (en este proyecto se incluye en JavaScript³³), que no es más que un guión que realiza diversas tareas como combinar plugins e interactuar con el sistema operativo o con el usuario.

Una de las características del lenguaje HTML que le ha aportado la popularidad y extensión de la que dispone, es que puede ser creado y editado por cualquier editor de texto básico o que admita texto sin formato.

Otra opción verdaderamente considerable son los denominados editores WYSIWYG (What You See Is What You Get) que permiten ver el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando la página web.

Esto no se traduce en una manera distinta de realizar la web ya que el código

³² Forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

³³ Lenguaje de programación interpretado que se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador, etc.

que se genera es el mismo que se podría generar en un editor de texto general, sino en una forma más simplificada de edición. Esto es debido a que estos programas, además de tener la opción de trabajar con la vista preliminar, tienen una sección HTML, la cual va generando el código a medida que se va trabajando.

Combinar ambos métodos de programación, editar el código a mano y editarlo con el segundo grupo de editores, resulta muy cómodo y visual, ya que de alguna manera, se complementan entre sí.

En este proyecto se utilizan ambas técnicas de programación web, utilizando Wordpad como editor de texto básico y Adobe Dreamweaver como editor WYSIWYG.

8.2.1. Marcado HTML

El estándar consta de varios componentes vitales, entre ellos los elementos y sus atributos. Los elementos son la estructura básica de HTML que tienen dos propiedades básicas: los atributos y el contenido.

Cada atributo y contenido tienen ciertas restricciones para que se considere válido para el documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo `<elemento1>`) y una etiqueta de cierre (siguiendo el ejemplo, `</elemento1>`).

Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido entre las dos etiquetas (por ejemplo, `<elemento1 atributo="valor"> Contenido </elemento1>`).

Se pueden definir tres tipos de marcados usuales en HTML:

- ❖ Marcado estructural: describe el propósito del texto. Las marcas son del tipo `<H#> Título </H#>` donde # es un número comprendido entre el 1 y el 6. También son denominados encabezados y se emplean para marcar los títulos en las secciones.
- ❖ Marcado presentacional: Describe la apariencia del texto sin importar su función. La mayoría del marcado presentacional ha sido desechada a partir del estándar HTML4.0.
- ❖ Marcado hipertextual: Se utiliza para enlazar partes del documento con otros documentos u otras partes del mismo documento. Para crear un enlace es necesario abrir la etiqueta de ancla `<a>` junto con el atributo `href`, que establecerá la dirección a la que apunta el enlace.

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo igual ("=") y escritos en la etiqueta de comienzo de un elemento,

después del nombre de este. El valor puede estar rodeado por comillas dobles o simples. Generalmente, cada etiqueta consta de uno o varios atributos.

El contenido es todo aquello comprendido entre la etiqueta de inicio (con sus atributos) y la etiqueta de cierre y al que se le aplica todo lo relacionado con la etiqueta de inicio.

8.2.2. Etiquetas básicas

Las etiquetas básicas que define el estándar HTML son:

- ❖ <HTML>: define el inicio de un documento HTML, indicando al navegador que el código que viene a continuación debe interpretarse como ese lenguaje.
- ❖ <script>: Incrusta un script en la página web.
- ❖ <head>: Define la cabecera del documento HTML, que suele contener información que no se muestra directamente al usuario. Dentro de la cabecera es posible encontrar:
 - <title>: define el título de la página.
 - <link>: para vincular el sitio a hojas de estilo³⁴ o iconos.
 - <style>: para colocar el estilo interno de la página. Si se vincula mediante link a un archivo externo no es imprescindible incluirlo.
 - <meta>: para metadatos.
- ❖ <body>: define el contenido principal del documento, siendo la parte que se muestra en el navegador y donde se pueden definir propiedades comunes a toda la página. Dentro del cuerpo es posible encontrar:
 - <table>: crea una tabla
 - <a>: hipervínculo dentro o fuera de la propia web.
 - <div>: es un contenedor a nivel de bloque para otros elementos. Adquieren su potencial combinándolos junto con las hojas de estilos, detalladas en el apartado siguiente, ya que resultan muy útiles para asignar atributos presentacionales a bloques enteros de contenido.
 - : Indica la inserción de una imagen en la web.
 - : las listas se definen mediante y y cada ítem que queramos incluir se debe hacer mediante la etiqueta . La diferencia entre y es que mientras la primera son listas no ordenadas y la segunda son ordenadas.
 - <form>: inserta un formulario.

³⁴ Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML.

8.2.3. Hojas de estilos en cascada o CSS

Las hojas de estilos en cascada o CSS juegan un papel fundamental en la presentación de la web ya que separan la estructura de un documento de su presentación.

La información del estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. Para ello, pueden definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante la etiqueta `<style>`.

CSS tiene una sintaxis muy sencilla, que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos.

Una hoja de estilos CSS consiste en una serie de reglas: cada regla consiste en uno o más selectores y un bloque de estilos con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato "propiedad:valor;".

En el CSS, los selectores marcarán qué elementos se verán afectados por cada bloque de estilo que les siga, pudiendo afectar a uno o varios elementos a la vez, en función de su tipo, nombre (name), ID, clase (class), etc.

8.3. Estándar HTML5

Es la última versión de HTML, caracterizada por establecer una serie de nuevos elementos y atributos que reflejan el uso típico de las páginas webs modernas.

Algunos de ellos son técnicamente similares a etiquetas tales como `<div>` pero tienen un significado semántico, como por ejemplo `<nav>` (bloque de navegación de la web) y `<footer>` (pie de página).

Incorpora además nuevas etiquetas como `<audio>` o `<video>`, que permiten la opción de publicar contenido multimedia sin la necesidad de incluir un plugin o complemento.

Finalmente, incluye también mejoras en la etiqueta `<canvas>`, capaz de renderizar en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE) elementos 3D.

8.4. Desarrollo

Para la realización de la aplicación HTML5 del sistema del que es objeto el presente documento, se utiliza el Software de programación WYSIWYG Adobe Dreamweaver.

La aplicación consta de una página web adaptada para formato PC y otra para formato móvil donde se podrá visualizar el contenido emitido en directo y el contenido suministrado bajo demanda por el servidor.

Aunque la idea inicial era utilizar, tanto para el Streaming en directo como para el Streaming Bajo Demanda el plugin de VLC, debió descartarse esa opción debido a la inexistencia de plugin de VLC para dispositivos móviles Android.

Así pues, se optó por utilizar el complemento de VLC para el visionado de flujo en directo a través de la web del ordenador, y las etiquetas específicas multimedia que HTML5 proporciona en su mejora del estándar para el contenido Bajo Demanda.

Para la web adaptada a dispositivos móviles, se mantienen las etiquetas multimedia de HTML5 para los videos Bajo Demanda, pero el flujo de vídeo en directo se realiza a través del Software de VLC para Android, vinculado a la propia página web.

CAPÍTULO 9. Android

9.1. Introducción

En los inicios de la telefonía móvil, los terminales móviles tenían unas funcionalidades muy limitadas que solo permitían al usuario realizar aquellas funciones que venían instaladas de fábrica. Aunque cada terminal era distinto, ya que dependía del Software utilizado por cada fabricante, todos tenían funciones más o menos comunes.

Con el paso de los años y el avance de las tecnologías, los teléfonos móviles fueron permitiendo progresivamente a los usuarios instalar pequeñas aplicaciones, sobre todo juegos Java, aunque como el sistema operativo del móvil aún dependía del fabricante, el usuario debía asegurarse de que aquello que pretendía instalar era compatible con su terminal.

Más tarde, debido a la aparición de nuevos y evolucionados dispositivos en el mercado, se hizo indispensable el poder instalar y desinstalar aplicaciones más complejas acordes a las nuevas necesidades del mercado y los usuarios.

Puesto que no era rentable ni eficiente el tener que crear una misma aplicación con funcionalidades idénticas para el sistema operativo de cada fabricante, se desarrollaron nuevos sistemas operativos compatibles con los terminales móviles de distintos fabricantes.

Los primeros de estos nuevos sistemas operativos para teléfonos móviles fueron Symbian (de Panasonic, Siemens, AG, Nokia y Sony-Ericsson, entre otras), Palm (diseñado para PDA's), Blackberry y Windows Mobile (de Microsoft).

Años más tarde, en junio de 2007, Apple lanzó a la venta su familia iPhone, caracterizada por su sistema operativo iPhone OS (iOS) y su pantalla multitáctil, revolucionando así el mundo de la telefonía móvil conocido hasta entonces.

Aproximadamente un año después, el nuevo sistema operativo Android se dio a conocer al mundo de la mano de Google, provocando una nueva revolución del mercado debido a su potencial y a su versatilidad.

9.2. Historia

Android es un sistema operativo, inicialmente desarrollado para teléfonos móviles por Android Inc, una compañía de Software poco conocida ubicada en California.

Aunque Android era un sistema operativo prácticamente desconocido, en 2005 Google formalizó su primer paso hacia el mundo de la telefonía móvil comprando dicha compañía y provocando de esta forma el creciente interés del mundo por el sistema operativo.

En noviembre de 2007 se creó la Open Handset Alliance, un consorcio de varias compañías de Hardware, Software y Telecomunicaciones que agrupaba a varios fabricantes, y se lanzó la primera versión de Android junto con su respectivo SDK (Software Kit Developer), con el objetivo de que los programadores empezaran a crear aplicaciones para este sistema.

Aunque los inicios fueron un poco lentos, debido a que se lanzó antes el sistema operativo que el primer móvil Android, actualmente es uno de los sistemas operativos multidispositivo más vendido en el mundo.

9.3. Características

Android es un sistema operativo para móviles muy avanzado basado en Linux, un núcleo de sistema operativo, libre, gratuito y multiplataforma. Aunque inicialmente fue pensado únicamente para teléfonos móviles o smartphones, al igual que iOS, Symbian y Blackberry, actualmente también es compatible para varios dispositivos, entre ellos, tablets.

A diferencia del resto de sistemas operativos mencionados, Android permite desarrollar y programar aplicaciones de una forma muy sencilla, en un lenguaje de programación que es una variación de Java llamada Dalvik, proporcionando todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.).

El hecho de realizar las aplicaciones en Java aporta a los terminales más versatilidad, pues les permite utilizar aplicaciones diseñadas para otros sistemas operativos pero que también usan Java como plataforma. Además, cuentan con un navegador basado en Google Chrome y el soporte para Hardware es muy amplio: desde dispositivos USB, inalámbricos y hasta pantallas táctiles de gran tamaño y resolución.

En la imagen siguiente puede verse el diagrama de arquitectura del sistema operativo de Android:

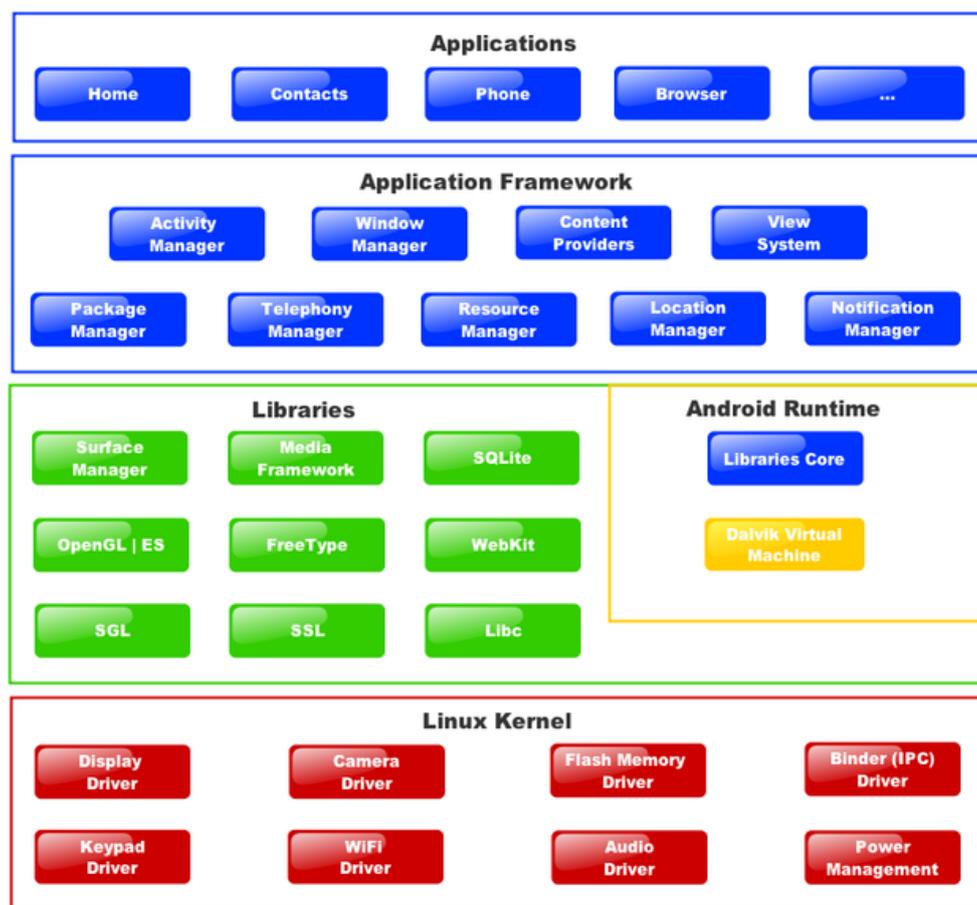


Fig. 9.1 Diagrama de arquitectura de Android

Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hace que una de las características más importantes y diferenciales de este sistema operativo sea la cantidad de aplicaciones disponibles, debido a que cualquier persona con una mínima noción de Java puede desarrollar las suyas propias o mejorar las ya existentes, de forma que el usuario puede tener un teléfono hecho a medida.

De esta manera, se evita que se generen cientos de componentes de aplicaciones distintas que respondan a la misma acción, dando la posibilidad de que los programas sean modificados o reemplazados por cualquier usuario sin tener que empezar a programar sus aplicaciones desde el principio.

Otra de las características más relevantes de este sistema operativo es que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono, hay que pagar nada, hecho que lo hace muy popular entre fabricantes y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos.

Cualquiera puede bajarse el código fuente, inspeccionarlo, compilarlo e incluso cambiarlo. Esta característica da seguridad a los usuarios, ya que algo que es abierto permite detectar fallos más rápidamente y así repararlos. También es

ventajoso para los fabricantes, ya que pueden adaptar mejor el sistema operativo a los terminales.

Actualmente, Android soporta una gran variedad de formatos multimedia como WebM, H.263, H.264, MPEG-4 SP, AAC, MP3, Ogg Vorbis y WAV entre otros. También soporta la transmisión mediante Streaming RTP/RTSP, la descarga progresiva de HTML (HTML5 <video> tag) y el Adobe Flash Streaming (RTMP), si se dispone de Adobe Flash Player.

Desde el punto de vista técnico, Android usaba un núcleo de Linux debido a varias razones: no había que pagar derechos a otras compañías, el sistema de seguridad para el usuario es muy superior en todos los aspectos, los núcleos Linux son muy versátiles ya que se pueden compilar prácticamente para todo tipo de procesadores y cuenta con una plataforma de desarrolladores (voluntaria) muy amplia.

El núcleo de Android, por ser un núcleo Linux, puede compilarse a voluntad de cada fabricante ya que es código abierto o libre. En el caso de otros sistemas operativos de código propietario, la compilación solo puede ser realizada por el dueño del Software, factor que implica ceñirse a las condiciones y requisitos de éste, así como aumentar el coste en sí.

Como resultado de esto cada fabricante hace las modificaciones necesarias al sistema operativo sin tener que modificar su Hardware y como las aplicaciones se realizan en la plataforma de Java, se pueden usar en cualquier dispositivo que tenga Java instalado.

En la actualidad un porcentaje muy grande de los terminales móviles usan Android como plataforma y al parecer su crecimiento va en aumento (prácticamente de forma exponencial) gracias a estas características.

9.4. Actualizaciones y versiones

El historial de versiones del sistema operativo Android se inició con el lanzamiento de Android Beta en noviembre del 2007. La primera versión comercial, Android 1.0, fue lanzada en septiembre del 2008.

Las actualizaciones a su sistema operativo, desarrolladas bajo un nombre en clave y lanzadas en orden alfabético, típicamente corrigen fallos de programa y agregan nuevas funcionalidades.

Versiones	Características
Beta	• Fue lanzada el 5 de noviembre del 2007, mientras que el SDK fue lanzado el 12 de noviembre de 2007.
v.1.0	• Fue la primera versión comercial del software.
Apple Pie	• Fue lanzada el 23 de septiembre del 2008.

	<ul style="list-style-type: none"> • Agregó una serie de características entre las que destaca la inclusión de un navegador web para visualizar páginas web HTML y XHTML.
v.1.1 Banana Bread	<ul style="list-style-type: none"> • Fue lanzada el 9 de febrero del 2009. • La actualización resolvió fallos, cambió la API y agregó una serie de características adicionales, entre las que destaca la posibilidad de guardar los archivos adjuntos en los mensajes.
v.1.5 Cupcake	<ul style="list-style-type: none"> • Fue lanzada el 30 de abril de 2009. • Está basada en un núcleo Linux 2.6.27. • Incluye correcciones de interfaz de usuario así como varias características adicionales entre las que destaca la posibilidad de grabar y reproducir en formatos MPEG-4 y 3GP.
v.1.6 Donut	<ul style="list-style-type: none"> • Fue lanzada el 15 de septiembre del 2009 junto con el SDK de Android 1.6. • Está basada en el núcleo de Linux 2.6.29. • Incluye numerosas características nuevas entre las que destaca la incorporación de un motor multi-lenguaje de Síntesis para el habla (Text to Speech).
v.2.0/2.1 Éclair	<ul style="list-style-type: none"> • Fue lanzada el 26 de octubre del 2009, junto con el SDK 2.0. • Está basada en un núcleo Linux 2.6.29. • Incluye correcciones de errores y múltiples características adicionales entre las que destaca el soporte para HTML5.
v.2.2 Froyo	<ul style="list-style-type: none"> • Fue lanzada el 20 de mayo del 2010 junto con el SDK de Android 2.2. • Está basada en el núcleo de Linux 2.6.32. • Incluye correcciones de errores así como numerosas características nuevas entre las que destacan las mejoras en velocidad, memoria y rendimiento y el soporte para Adobe Flash.
v.2.3 Gingerbread	<ul style="list-style-type: none"> • Fue lanzada el 6 de diciembre del 2010, junto con el SDK 2.3. • Está basada en un núcleo Linux 2.6.35. • Incluye correcciones de errores y múltiples características adicionales entre las que destaca el soporte para reproducción de vídeo por WebM/VP8 y encodeo de audio por AAC.
v.3.0/3.1/3.2 Honeycomb	<ul style="list-style-type: none"> • Fue lanzada el 22 de febrero del 2011 junto con el SDK de Android 3.0. • Fue la primera actualización única para tablet Android. • Está basada en el núcleo de Linux 2.6.36. • Incluye numerosas características nuevas entre las que destaca el soporte optimizado para tablets con una nueva y "virtual" interfaz de usuario holográfica.
v.4.0 Ice Cream Sandwich	<ul style="list-style-type: none"> • Fue lanzada el 19 de octubre del 2011, junto con el SDK 4.0.1. • Está basada en un núcleo Linux 3.0.1. • El código fuente para Android 4.0 se puso a disposición el 14 de noviembre del 2011.

	<ul style="list-style-type: none"> • Incluye correcciones de errores y múltiples características adicionales entre las que destaca la incorporación de un nuevo navegador web bajo la marca de Google Chrome permitiendo hasta 15 pestañas y la unificación de los dos sistemas (teléfonos móviles y tablets) en uno solo.
V4.1/4.1.2/4.2 Jelly Bean	<ul style="list-style-type: none"> • Fue lanzada el 27 de junio del 2012. • Está basada en el núcleo de Linux 3.0.31. • Incluye numerosas características nuevas entre las que destaca la mejora de la funcionalidad y rendimiento de la interfaz de usuario.

Tabla 9.1. Versiones y actualizaciones de Android

9.5. Desarrollo

Para poder desarrollar y programar la aplicación con ayuda de la cual se mostrará el funcionamiento multidispositivo del proyecto del que es objeto el presente documento, se descarga el SDK de Android, con todas las librerías necesarias y el emulador, desde su página web oficial.

Los SDK incluyen un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del Software. En entorno de desarrollo se integra en la plataforma libre Eclipse.

La aplicación permitirá la visualización del contenido multimedia, alojado en el servidor que forma parte del proyecto, mediante transmisión Streaming tanto Live como Bajo Demanda.

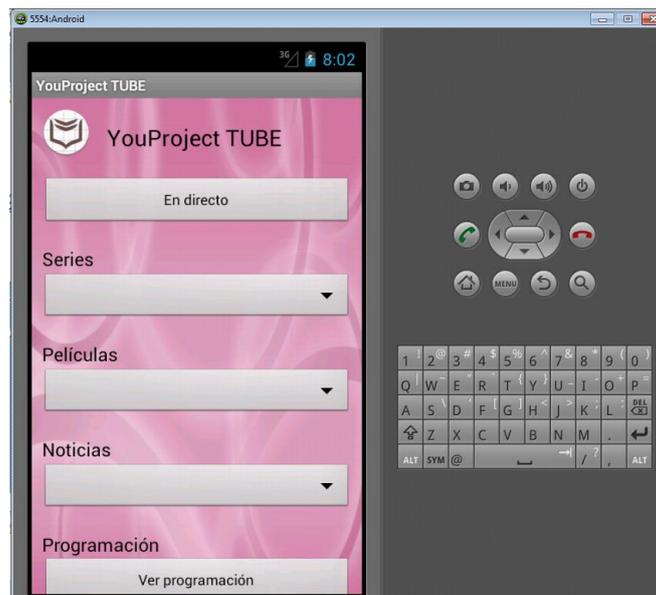


Fig. 9.2 Aplicación en el simulador

CAPÍTULO 10. Presupuesto

10.1. Introducción

En este capítulo se detalla todo lo relacionado con los costes que serían necesarios para la implementación de este proyecto. Aunque el sistema de pruebas diseñado para la validación de dicho sistema es gratuito, ya que se reutilizan ordenadores personales y software libre para la implementación de los distintos módulos que lo componen, se detallará de igual forma en este capítulo.

10.2. Presupuesto del sistema de pruebas

Para el denominado sistema de pruebas, se expondrá la opción más económica que permita la correcta validación del funcionamiento del proyecto, reutilizando elementos de uso personal y software libre y gratuito.

Presupuesto del sistema de pruebas				
1.INVERSIÓN				
Concepto	Elemento	Precio unitario	Cantidad	Precio total
Hardware	Ordenador con webcam integrada	799€	1	799€
Hardware	Ordenador	799€	1	799€
Hardware	Dispositivo móvil Android	215€	1	215€
Software	Ubuntu	0€	1	0€
Software	VLC Media Player	0€	2	0€
Instalación	N/A	N/A	N/A	N/A
2.INGENIERÍA				
Concepto	Elemento	Precio unitario	Cantidad	Precio total
Dirección proyecto	Incluye documentación, dirección y pruebas para la validación del funcionamiento del sistema.	N/A	N/A	N/A
Dietas	N/A	N/A	N/A	N/A
Desplazamientos	N/A	N/A	N/A	N/A
3.MANTENIMIENTO				
Concepto	Elemento	Precio unitario	Cantidad	Precio total
Mantenimiento	N/A	N/A	N/A	N/A

4.COSTE TOTAL	
Coste total	1.813€

Fig. 10.1 Presupuesto del sistema de pruebas

- Ordenador “cliente”: simulará ser el cliente que emitirá la señal que verán los usuarios. Deberá disponer de una webcam integrada (o bien una configurada y conectada mediante USB) y tener el software VLC Media Player para la transmisión de la señal de vídeo, simulando el reproductor del cliente y el bloque de codificación respectivamente.

En este caso, y dado el objetivo y ámbito de este sistema, se utilizará el mismo ordenador “cliente” como “servidor”, aunque podrían utilizarse dos ordenadores diferentes.

- Ordenador “usuario final”: simulará ser el usuario final que reciba el contenido tanto live como bajo demanda transmitido por el “cliente”. No requerirá de la instalación de ningún software ni hardware específico para la visualización de dicho contenido.

Se dispondrá también de un dispositivo móvil Android que será utilizado para la validación de la transmisión Streaming multidispositivo del sistema, así como de la aplicación propuesta.

CAPÍTULO 11. CONCLUSIONES

El objetivo principal del proyecto del que es objeto el presente documento, era conocer el funcionamiento de la transmisión Streaming así como los elementos y tecnologías necesarias para llevarla a cabo de la forma más optimizada posible.

Inicialmente se propuso un sistema puramente teórico debido al elevado coste económico que requiere la implementación completa y real de este tipo de sistemas, que puede verse reflejado, por ejemplo, en el coste de los elementos de codificación.

Puesto que de esta forma no podía validarse el funcionamiento completo del sistema en sí, se optó finalmente por adoptar una solución de coste cero con una arquitectura más modesta, si la comparamos con la de un sistema comercial, pero que permitía comprobar y mostrar su funcionamiento completo de una forma sencilla e ilustrativa.

Así pues, se optó por utilizar la señal de una webcam de ordenador como la señal Live que podría emitir un radiodifusor cualquiera, ordenadores comunes para simular los equipos de emisión y recepción, así como el servidor físico que estaría (generalmente) alojado en las instalaciones del cliente y software libres y gratuitos que harían las funciones realizadas por los elementos de codificación que formarían parte indispensable del sistema.

Aunque fue la alternativa tomada para poder conseguir mostrar su funcionamiento y viabilidad, se decidió no perder la referencia aportada por el punto de vista comercial a lo largo del proyecto y su posible exportación a tales ámbitos, con el fin de no limitarlo a un mero uso casero o didáctico, sino de poder ver sus capacidades e implementación en el caso de disponer de los elementos y recursos necesarios para ello.

Por otro lado, y con el objetivo de cumplimentar lo comentado en este capítulo y a lo largo del presente documento, se decidió estudiar de forma meramente teórica, debido nuevamente a las limitaciones económicas y su consecuente indisponibilidad de recursos necesarios, así como la inexistencia de soluciones gratuitas alternativas, formas de aportar ventajas adicionales al sistema.

En el caso particular de este proyecto, se decidió optar por la inclusión de determinados elementos que aportaban valor añadido al sistema, complicando mínimamente su arquitectura y aportando funcionalidades consideradas, bajo nuestro punto de vista, lo suficientemente importantes y significativas como para ser dignas de mención.

Entre ellas están la inclusión de CDNs, con el objetivo de permitir una accesibilidad más rápida a los contenidos por parte de los usuarios finales, ya que al fin y al cabo es un requisito fundamental cuando hablamos de transmisión mediante Streaming.

Otra de las también contempladas es la inclusión de N servidores web balanceados, pensado para aquellos sistemas con un gran volumen de usuarios y su consecuente volumen de peticiones a atender, favoreciendo así la velocidad de respuesta por parte del sistema y disminuyendo la posibilidad de colapso del sistema.

Por último, y no por ello menos importante, se plantea la inclusión de codificadores que permitan la redundancia de estos mismos, aumentando de esta forma la robustez del sistema o bien aporten funcionalidades adicionales como la escalabilidad o la posibilidad de convertir la señal a formatos diferentes, con el objetivo de adaptarse de una forma más óptima a las nuevas necesidades del mercado en lo que a transmisión multidispositivo se refiere.

Puesto que el proyecto del que es objeto el presente documento se basa prácticamente de forma única en componentes Software, se considera que no tiene impacto medioambiental alguno.

A nivel personal, este trabajo nos ha permitido conocer mejor el funcionamiento de los sistemas de transmisión mediante Streaming y nos ha aportado una visión más amplia y detallada sobre conceptos de compresión, codificación y transporte de las señales.

Por otra parte, también nos ha hecho estudiar, de todas las modificaciones posibles, aquellas que serían más interesantes de aplicar sobre el sistema del que es objeto el presente documento, en caso de que decidiera realizarse una implementación a nivel comercial, valorando la funcionalidad adicional que aportaría así como la complejidad y el incremento de coste que estas mismas comportarían.

BIBLIOGRAFÍA

- [1] Barceló, J.M., Íñigo, J., Llorente, S., Marquès i Puig, J.M., Martí, R., Peig, E. y Perramon, X., "Protocolos y aplicaciones Internet", Editorial UOC (2008).
- [2] Perkins, C., "RTP Audio and Video for the Internet", Addison-Wesley (2003).
- [3] Dai, N., Mandel, L. and Ryman, A., "eclipse Web Tools Platform. Developing Java™ Web Applications", Addison-Wesley (2007).
- [4] Blum, R., "Linux® Command Line and Shell Scripting Bible", Wiley Publishing, Inc. (2008).
- [5] Hill, B.M., Bacon, J., Burger, C., Jesse, J. and Krstić, I., "El libro oficial de Ubuntu", Anaya Multimedia (2007).
- [6] Ghanbari, M., "Standard Codecs: Image Compression to Advanced Video Coding", IEE Telecommunications Series 49 (2003).
- [7] Black, U., "TCP/IP and Related Protocols", McGraw-Hill (1998).
- [8] Rovira, D., "Dreamweaver CS5: curso de iniciación", Barcelona Inforbooks (2011).
- [9] hyc, "IPTV Course", 11 sessions (2007).
- [10] Marqués, F., Vidal, J. and Bonafonte, A. "Coding and Transmission of Multimedia Contents" (2006).
- [11] Cisco, "Cisco Digital Media Suite: Cisco Digital Media Encoder 2000", Data Sheet (2010).
- [12] Department of Media Technologies (La Salle – Universitat Ramon Llull), Scalable Video Coding: MPEG-4 part 10 - SVC (2010).
- [13] Moreno, L., "Curso Nuevas Tecnologías Aplicables" (2011).
- [14] Campus, "Curso WebTV" (2007).
- [15] Furgusson, C., "MPEG-4 AVC advances throughout the network", TVBEurope HD Europe (2008).
- [16] Fernández, P., Fuentes, R. and Ramírez F.A., "Propagación del Error en Codificación de Vídeo Escalable" (2011).
- [17] Moreno, A., "Codificación digital" (2010).
- [18] Albacete, A., "Gestión completa de vídeo", ISID Media (2002).

- [19] Xpertia Soluciones Integrales, “Estado del Arte de las Tecnologías Audiovisuales. De la captación de imágenes al usuario final. Actualización del 2012 (2012).
- [20] Morales, N., “Streaming Live con Windows Media Encoder”, Universidad Tecnológica de Panamá (2010).
- [21] Suárez, F.J., “Tecnologías de Streaming”, Universidad de Oviedo – Área de Arquitectura y Tecnología de Computadores (2011).
- [22] Universidad Politécnica de Vic, “Transmisión en Internet: Streaming de audio y vídeo (2012).
- [23] Lanza, J.M., García, M.J. and Ruiz, J.C., “Lógica y computabilidad: Codificación”, Universidad de Extremadura (2009).
- [24] Ozer, J., “Scalable Video Coding: The future of Video Delivery?” (2009).
- [25] Rijo, D., “Fundamentos de Video Streaming”, Universidad de la República Montevideo – Uruguay (2004).
- [26] Ochoa-Domínguez, H.J, Mireles-García, J. and Cota-Ruiz, J.D., “Descripción del nuevo estándar de vídeo H.264 y comparación de su eficiencia de codificación con otros estándares”, Departamento de Ingeniería Eléctrica y Computación – Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez (2006).
- [27] Ruiz, R., “El H.264 un nuevo estándar para compresión de vídeo y su aplicación a la videoconferencia” (2010).
- [28] rnds, “Estándar de compresión de vídeo H.264”.
- [29] Krohn, A., “10 pasos fáciles para usar Google App Engine como tu propia red de distribución de contenido” (2008).
- [30] Salavert, A., “Formatos de vídeo digital”.
- [31] Colmenar, A. “Tecnología Multimedia: Vídeo Digital”.
- [32] López, M.F., Rodríguez, S.G., Ortiz, J.P., Ruiz, V.G. and García, I. “Técnicas para la Codificación Escalable de Vídeo”.
- [33] Vilchez, A., “Qué es Android: Características y Aplicaciones” (2009).
- [34]<http://www.sonymobile.com/es/support/faq/xperia-s/apps-android-market/what-is-android-1/>
- [35]<http://www.neoteo.com/google-lanza-el-formato-abierto-webm>
- [36]<http://hernaaan.com.ar/wiki/vorbis.html>

- [37]<http://www.fotonostra.com/digital/conperdida.htm>
- [38]<http://www.xatakaon.com/tecnologia-de-redes/cdn-que-es-para-que-sirve-y-por-que-no-rompe-con-la-neutralidad-de-la-red>
- [39]<http://www.slideshare.net/bernycol/h264-codec-multimedia#btnNext>
- [40]<http://www.sonystreaming.com/video-en-directo/>
- [41]<http://grupos.cl/codificacion-escalable/>
- [42]http://videoconferencia.datalux.es/banners_marcas.php?id=3
- [43]http://www.taringa.net/posts/videos/7588726/Tipos-de-Streaming_-_que-podemos-transmitir-y-como_.html
- [44]<http://medialab-prado.es/article/streaming>
- [45]<http://www.desarrolloweb.com/articulos/482.php>
- [46]<http://www.consumer.es/web/es/tecnologia/internet/2009/09/18/187866.php>