

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

I.T. TELECOMUNICACIÓN (SIST. DE TELECOMUNICACIÓN)



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

“Sistema de captura de imágenes de streaming para el mantenimiento de la continuidad de la señal de TV(UPV-TV)”

TRABAJO FINAL DE CARRERA

Autor/es:
Ghita Ibnoulkhatib

Director/es:
Ignacio Boisch Roig

GANDIA, 2011



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

**Ingeniería Técnica de Telecomunicación
especialidad en Sistemas de Telecomunicación**

PROYECTO FINAL DE CARRERA:

“Sistema de captura de imágenes de streaming para el mantenimiento de la continuidad de la señal de TV en la UPV-TV”

IBNOULKHATIB Ghita

Índice

1.	RESUMEN:	4
2.	INTRODUCCIÓN:	5
2.1	La transmisión de la señal de Televisión: el comienzo de la historia	5
2.2	La transmisión de Televisión por Satélite:	8
2.3	La transmisión de Televisión por Cable:	9
2.4	La Televisión Digital:	10
2.5	La Televisión en Internet:	11
2.6	La evolución de las redes de datos y de Televisión:	12
3.	INTRODUCCIÓN A LOS SISTEMAS DE STREAMING:	14
3.1	Contenidos Multimedia:	15
3.2	Codificación de Video:	16
3.3	CODEC's utilizados por los Sistemas de Streaming:	18
3.4	Tipos de servicios de Streaming:	21
3.4.1	En directo (live):	21
3.4.2	Bajo demanda(on-demand)	22
3.4.3	Casi bajo demanda:	23
3.5	Componentes de un sistema de Streaming	23
3.5.1	Servidor de Streaming:	23
3.5.2	Red de comunicación:	24
3.5.3	Clientes:	25
3.6	Requisitos de un Sistema de Streaming.	27
3.7	Formatos de Archivos de Streaming.	29
3.7.1	Metafiles – Meta archivos:	30
3.7.2	Archivos de Audio:	31
3.7.3	Archivos de Video:	32
4.	POLÍTICAS DE GESTIÓN DE LOS SERVIDORES DE STREAMING:	34
4.1	Control de Recursos:	35
4.2	Planificadores:	35
4.3	Políticas de servicio:	37
4.4	. Sistema de almacenamiento:	40
5.	ARQUITECTURAS UTILIZADAS EN LOS SISTEMAS DE STREAMING:	41
5.1	Arquitecturas centralizadas:	41
5.2	Arquitecturas de servidores independientes:	44

5.3	Arquitecturas basadas en servidores-proxy:.....	45
5.4	Arquitecturas distribuidas a nivel de los usuarios:	48
6.	ARQUITECTURAS DE STREAMING A GRAN ESCALA:	50
6.1	Requisitos de las arquitecturas de Streaming a gran escala:.....	50
6.1.1	Escalabilidad del Sistema de Streaming:	50
6.1.2	Tolerancia a fallos:.....	51
6.1.3	Costo del Sistema de Streaming:.....	52
6.1.4	Balanceo de la Carga:	53
6.1.5	Compartición de Recursos:	53
6.2	Alternativas actuales para los sistemas de Streaming a Gran Escala:	54
7.	ESTRUCTURA DE UN CANAL DE TELEVISION.....	57
7.1	Proceso de Transmisión de un Canal de Televisión.....	58
7.1.1	Modelo de Transmisión General	58
7.1.2	Modelo de Transmisión Reducido.....	61
8.	LEER UN STREAM DE UN VIDEO FORMATO AVI DIVIDIRLO EN FRAMES.....	64
8.1	Breve reseña histórica del formato AVI:.....	64
8.2	Cómo funciona el formato AVI:.....	64
8.3	Cómo se reproduce un archivo AVI.....	65
8.4	Código en Visual Studio 6.0 de extracción de un video Avi y dividirlo en Frames:.....	66

1. RESUMEN:

Hoy en día es más común que las personas estén interesadas en recibir información por Internet de la mejor forma posible, esto quiere decir, de forma clara y entendible. Esto se debe al gran crecimiento de Internet en los hogares, y de que los precios de conexión han bajado considerablemente en los últimos años. Es por ello que la transmisión de un canal de televisión por Internet se hace hoy en día indispensable para la difusión de información a todas las personas de forma eficiente y clara.

Como dicha transmisión en vivo necesita que la información llegue lo más rápido posible a los usuarios, se creó la tecnología de streaming, que no es más que la transmisión de archivos multimedia en tiempo real, los cuales pueden ser reproducidos a medida que llegan los paquetes a los usuarios, por lo que no deben ser descargados por completo para realizar dicha acción.

Para poder transmitir los contenidos multimedia, se necesita un servidor de streaming que se encarga de enviar la información a los usuarios.

2. INTRODUCCIÓN:

2.1 La transmisión de la señal de Televisión: el comienzo de la historia

Desde los tiempos de la prehistoria de la televisión, hacia finales del siglo XIX, se ha buscado la forma de transmitir imágenes (que no difundir, todavía), de la misma forma que se transmitía ya sonido a través de la radio.

En los 20's aparecieron los dos primeros modelos: la televisión mecánica (John Baird 1.929 con BBC) y la electrónica (Vladimir Zworykin 1.931 con RCA). Dada la superior calidad de la televisión electrónica, una demostración de la compañía EMI a la BBC de Londres determinó el final del invento de John Baird. En 1.936, la BBC comenzó sus transmisiones desde los estudios Alexandra Palace, en Londres. RCA transmitía desde 1.931 desde el Empire State, de forma experimental.



Figura 2.1: la televisión mecánica (John Baird 1.929 con BBC)



Figura 2.2: la televisión la electrónica (Vladimir Zworykin 1.931 con RCA).

Es en esta década, la de los 30's, cuando los países tecnológicamente más avanzados comienzan sus emisiones experimentales, cada una con sus especificaciones técnicas: no existía ninguna normativa al respecto. En 1.941, la FCC (siglas en inglés de “*Federal Communications Commission*”) autorizó a la NBC (siglas en inglés de “*National Broadcasting Company*”, subsidiaria de RCA) la autorización para emitir, dando la bandera de salida a la radiodifusión de la televisión comercial, aunque las emisiones comenzaron en 1.939. El sistema adoptado de facto fue el de 525 líneas.

El resto de países más desarrollados comenzaron sus emisiones experimentales por esta época: Alemania con los JJ.OO. de Berlín de 1.936 y con 441 líneas, Italia con el sistema Alemán de 441 líneas, la URSS con 343 líneas, etc.

Sin embargo, el desarrollo de la televisión estaba lastrado por la ausencia de aparatos receptores comerciales. Fue en 1.937 y 1.939 cuando comenzaron las primeras fabricaciones en serie, en Gran Bretaña y EE.UU. respectivamente.

El comienzo de la 2º Guerra Mundial paralizó el desarrollo de la televisión. Tras su finalización, gobiernos y sectores industriales ligados a la televisión vieron en ésta un medio para la recuperación, como herramienta fundamental para el desarrollo social. Los servicios de televisión comenzaron a regularse y normalizarse. En 1.947 estableció la normalización técnica de todas las emisiones televisivas, lo que provocó

el reinició de la expansión del medio en EEUU.

La expansión de la televisión estableció un nuevo equilibrio con el resto de medios de comunicación de masas (prensa, cine y radio), cobrando la importancia que merecía. También surgieron dos formas de hacer el negocio de la televisión: en EE.UU. e Iberoamérica la industria televisiva se asentó en redes de **empresas privadas** y comerciales en competencia (cuyas cabeceras eran la ABC, la NBC y la CBS); en la Europa del Oeste de posguerra la reconstrucción implicó la construcción de sistemas públicos y nacionales de radio y televisión.

Los **50s** han sido calificados por muchos analistas como aquellos del “gran **salto de la televisión** en el mundo”, puesto que es entonces cuando los **servicios regulares** de televisión se extendieron gradualmente por las grandes urbes del mundo (aunque las coberturas geográficas eran muy limitadas). En España, *Phillips* y *RCA* organizaron en 1.948 exhibiciones de televisión en Barcelona y Madrid respectivamente. Las emisiones regulares comenzaron en 1.956. EE.UU. donde destacó una poderosa industria televisiva (en 1.952 unas 108 emisoras estadounidenses alimentaban las pantallas de unos 21 millones de hogares).

En **Japón**, la televisión pública JBC (siglas en inglés de “*Japan Broadcasting Corporation*”), comenzó a operar en **1.953** y al año siguiente hizo lo propio la primera estación comercial.

En los 50s, los avances técnicos pusieron color a la televisión (con la gran dificultad de hacerlo manteniendo la compatibilidad con los ya millones de receptores vendidos), y también mayor desconcierto. EE.UU. aprobó el sistema NTSC en 1.953. Francia, por su cuenta desarrolló en 1.959 el SECAM, que fue adoptado por el bloque del Este. La Alemana *Telefunken* desarrolló el sistema PAL en 1.963, y éste fue adoptado por el resto de los países europeos.

2.2 La transmisión de Televisión por Satélite:

La internacionalización de los programas es una tendencia observable a lo largo de la historia de la televisión.

Gran parte de los intercambios entre países y entre organismos internacionales se hacía con satélites de comunicaciones. En 1.962, el **Telstar I**, permitió la primera transmisión entre EE.UU. y Europa. Un año después, el **Syncom III** (primer satélite geostacionario), permitió a estadounidenses y europeos seguir los Juegos Olímpicos de Tokio de 1.964. Ante estos avances, los gobiernos vieron la necesidad de crear una red mundial de satélites. El **Early Bird** (Intelsat I), puesto en órbita en 1.965 por *Intelsat*, fue el primer satélite que tuvo esa finalidad.

También por esos años aparecieron los satélites de difusión directa (DBS, siglas en inglés de “*Direct Broadcast Satellite*”), que permitían la difusión de una decena de señales de televisión hasta su recepción en antenas parabólicas domésticas.

Hacia 1.989, la fusión de *Sky Channel* con el consorcio *British Satellite Broadcasting*, dio lugar al nacimiento de *BskyB* (hoy, Sky Digital), y dio el pistoletazo para la segunda generación de operadores televisivos vía satélite. En poco tiempo, la creación de diversas plataformas multicanal vía satélite en todos los países fue una realidad.

Un paso más se produjo cuando las plataformas satelitales se digitalizaron totalmente. Así, en EE.UU. se lanzó a mediados de 1.994 Direct TV. Mientras que en la UE *Canal Satellite Numérique* (filial de Canal Plus en Francia) se convertiría en la primera plataforma digital europea.

A finales del año 2.000, un informe auspiciado por Eutelsat indicaba que en los países de la UE se contaba con 15 millones de receptores de televisión por satélite; siendo los británicos (cinco millones), los franceses (tres millones), y los italianos y españoles (dos millones cada uno) sus principales usuarios.

2.3 La transmisión de Televisión por Cable:

Los comienzos de la televisión distribuida a través de sistemas de cable, tuvo su origen hacia finales de la década de 1.940 en EE.UU. El objetivo inicial era hacer llegar la televisión a zonas de sombra de radio.

Se trataba de operadores unidireccionales, emplazados en áreas rurales y ciudades pequeñas, que garantizan la recepción de algunas señales de televisión. En la década de los 70's las operadoras de cable competían con las de radio en condiciones de ventaja comercial, por lo que se las obligó a producir contenidos propios y a pagar derechos de transmisión de otras señales. Paulatinamente, la televisión por cable se extiende como alternativa de mayor cantidad y calidad de contenidos, pero de pago independiente.

En septiembre de 1.975, la **HBO** (siglas en inglés de “*Home Box Office*”) alquiló una conexión con el satélite **Satcom I**, lo que indirectamente permitió dejar de pensar en el cable como parte de la iniciativa local de televisión y comenzar a hablar de una “Nación cableada”. Mediante esta operación, HBO se colocó a la altura de las tres grandes cadenas estadounidenses y se convirtió en el modelo a seguir por otras compañías.

En Europa, la implantación del cable fue muy inferior, debido sobre todo al proteccionismo existente por las cadenas nacionales estatales. Holanda, Suiza y Bélgica fueron excepciones.

Desde mediados de los 80's, gracias a la digitalización, se permitieron nuevas formas comerciales de la televisión por cable, como el pago por visión, permitiendo cada vez mayor interactividad y opciones multimedia. La similitud de mercados y de opciones de contenido produjo una complementariedad o simbiosis entre operadoras de satélite y de cable.

Actualmente, en EE.UU. hay unos 73 millones de hogares abonados a empresas de televisión (un 70% de hogares); Japón cuenta con unos 15 millones de suscriptores (30% de hogares), mientras que en la UE son más de 50 millones los abonados, con

penetraciones en los respectivos mercados que van desde cifras cercanas o superiores al 90 por ciento en países como Bélgica, Holanda o Luxemburgo, a presencias meramente testimoniales de un 4 como España e Italia.

2.4 La Televisión Digital:

La generalización de los satélites de difusión directa (DBS) y de la televisión por cable (CATV) en vastas regiones del mundo, sumada a la digitalización de las señales y de los servicios ofrecidos por ambos soportes, ha producido un nuevo escenario televisivo con posibilidades no imaginadas años atrás: la multiplicación exponencial de señales, la fragmentación de audiencias y el surgimiento de nuevas formas comerciales (como el pago por visión). Muchos expertos consideran que, con la digitalización (producción, emisión y recepción de programas), la televisión tal y como la hemos conocido desaparecerá.

En apretada síntesis podemos decir que la Televisión Digital Terrestre permite, entre otras cosas:

- Incrementar notablemente el número de programas, servicios y señales actualmente disponibles.
- Mejorar la calidad de imágenes y de sonido de las transmisiones y de las recepciones televisivas.
- Establecer servicios personalizados e interactivos (relación empresa televisiva-audiencia) de radiodifusión y telecomunicaciones.
- Facilitar la convergencia entre el sector audiovisual, las telecomunicaciones y la informática

La migración ya ha comenzado en todos los países. El cambio está suponiendo un gran esfuerzo por parte de programadores, fabricantes de equipos receptores y operadores de redes, así como cuantiosas inversiones tanto de la industria como del público televidente.

Actualmente se encuentran en juego tres normas de transmisión de televisión digital

terrestre, elaboradas por empresas estadounidenses, japonesas y europeas. Estas normas - como ocurrió décadas atrás con los sistemas de televisión a color concebidos en EEUU, Francia y Alemania - compiten entre sí con la finalidad de captar la mayor cantidad de mercados posibles.

2.5 La Televisión en Internet:

Desde finales de la década de 1.990, asistimos a la creciente relación que establecen los sistemas de televisión digital en sus diferentes soportes (radio, satélite y cable) con la denominada red de redes: Internet. En el marco de esta relación, los televidentes se convierten en usuarios-consumidores que pagan distintos servicios interactivos (vídeo bajo demanda, videojuegos en red y tele-compra).

Si nos detenemos a observar que es lo que hoy está ocurriendo con la televisión en Internet, tendremos que miles de emisoras de televisión y productoras de todo el mundo se han volcado a crear sitios web en la Red. Por el momento, existe una gran variedad de tipos de sitios que van desde simples páginas institucionales donde, por ejemplo, se presenta información acerca de la empresa emisora y su programación (rejilla, sinopsis de programas, etc.) hasta sitios a partir de los cuales se puede acceder a la emisión en vivo a la emisión o a determinados programas de archivo. En estos últimos casos, un ancho de banda adecuado sumado a las tecnologías de “*streaming*” (transmisión de flujos de datos a través de Internet) de audio y vídeo, posibilitan la oferta de contenidos televisivos en la Red. Asimismo, entre el gran número de emisoras que emiten a través de Internet se encuentran las denominadas **bitcasters**, emisoras sólo existentes en la Red, que en muchas ocasiones se valen únicamente de webcams para difundir en directo los más variados contenidos.

En el marco de una Red que se ha ido expandiendo y comercializando rápidamente en los últimos años, las empresas de televisión y las operadoras de telecomunicaciones y operadoras de Internet se han ido relacionando a través de alianzas y fusiones: en 1.998 la cadena estadounidense *ABC* compró la mayoría de

Infoseek, y la *NBC* se alió con la empresa del magnate Bill Gates, *Microsoft*, conformando **MSNBC**.

Por otro lado, es posible observar cómo algunas televisiones o plataformas digitales de televisión enriquecen sus canales y servicios con elementos interactivos limitados, tomados o simulados del mundo Internet, utilizando a la Red como complemento y vía para mantener a los usuarios ante la televisión, por ejemplo los SMS.

Algunas empresas de televisión están ofertando una suerte de Intranet a la cual se accede a través del mando a distancia y la pantalla del televisor. **AOL** con más de 23 millones de abonados en Internet comenzó su trasvase a la televisión digital (1.500 hogares) a mediados de 2.002 ofertando servicios como el envío y recepción de correo electrónico o chats. En España, **Imagenio** (de Telefónica) permite la difusión de contenidos a través de una especie de Intranet que utiliza los mismos soportes y medios que al acceso de datos convencional.

Recientemente, empresas como **YouTube** y las nuevas redes sociales han convertido a los propios telespectadores en los productores de contenidos, lo cual supone lo último en esta revolución que vivimos en nuestra Sociedad de la Información.

2.6 La evolución de las redes de datos y de Televisión:

Las redes de datos existen desde los 60's, pero restringían su funcionalidad a la transmisión pura de datos entre ordenadores. El hecho de que la señal de audio y de vídeo no estuviera digitalizada hacía que la evolución de ambos fuera independiente.

Fue en la década de 1.990 cuando, con la digitalización de los contenidos de la televisión, tanto audio como vídeo, se empezó a considerar el uso de las redes de datos para transmitir estos contenidos.

Surgieron entonces problemas asociados a las características propias del audio y del vídeo. La gran cantidad de información que había que transmitir hizo por una parte que se fuera aumentando la capacidad de las redes con el tiempo, y por otra, estimuló el desarrollo de algoritmos de codificación que facilitaran esta transmisión.

Con la transmisión de audio y vídeo en tiempo real surgieron nuevos problemas, para los cuales las redes existentes no estaban preparadas, como el “*jitter*”, la pérdida de paquetes o el problema de la sincronización. Como consecuencia, nuevos protocolos, como el RTP (siglas en inglés de “*Real-time Transport Protocol*”), se han desarrollado para mitigar estos inconvenientes.

Las diferentes redes, soportes de transmisión, capacidades y características de los medios hace que las formas de codificación y transporte existentes constituyan un gran galimatías en ocasiones muy difícil de tratar. Salvo unas pocas más extendidas (MPEG, DVB, etc.), el número de normas es elevadísimo, y cada uno tiene la suya. Incluso existen ciertas incompatibilidades dentro de una misma norma.

Hoy en día se consideran las redes de datos como parte fundamental del sistema de transporte de audio y de vídeo, incluso adaptándose las características de las redes a las propias de la transmisión en tiempo real de audio y de vídeo (“*streaming*”).

3. INTRODUCCIÓN A LOS SISTEMAS DE STREAMING:

Streaming es una tecnología que permite la recepción instantánea, sin esperas, de información que fluye desde un servidor. Lógicamente ésta tecnología, que muchos pensarán que es de reciente aparición, está muy experimentada en el campo de Internet y surge de la necesidad de acceder a tipos de información voluminosa que generan amplios tiempos de espera usando la tradicional descarga de archivos. Esta información es, fundamentalmente, de tipo audiovisual aunque puede ser sólo audio (Radios en la Red) o vídeo (Canal de Televisión).

Desde su creación, los archivos de audio y vídeo han sido (y seguirán siendo a pesar de las compresiones) muy grandes. Los tamaños de los vídeos o de cualquier elemento multimedia pueden superar los MB y llegar a los GB lo que es impensable para un sistema que aún funciona demasiado lento como puede ser Internet. En definitiva, para concretar, muchos mega bytes para poder ser transmitidos con facilidad.

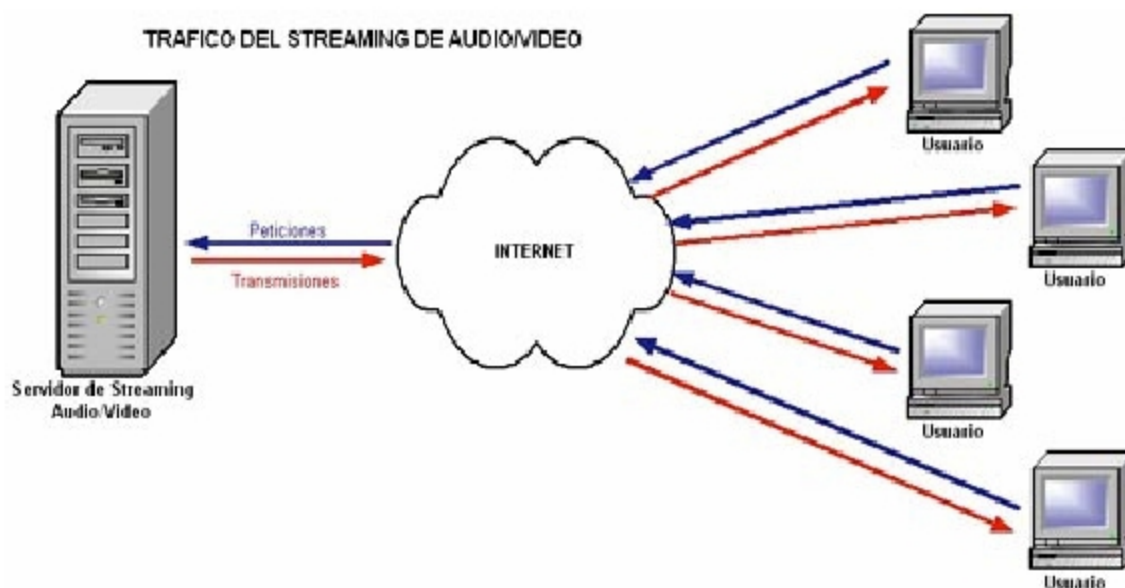


Figura 3.1: Comunicación entre el Servidor y los Clientes.

El término Streaming hace referencia a servicios en los cuales los usuarios son

capaces de pedir contenidos multimedia (videos) en cualquier instante de tiempo. Esta tecnología es de vital importancia para diversas aplicaciones multimedia como por ejemplo, aprendizaje a distancia, bibliotecas digitales, videoconferencias, Internet, televisión o sistemas de video bajo demanda.

3.1 Contenidos Multimedia:

La mayoría de la funcionalidad específica de los sistemas de Streaming deriva de las características particulares del tipo de información (contenidos multimedia) gestionada por estos sistemas. A diferencia de los tipos de datos tradicionales, los contenidos multimedia tienen una dimensión temporal explícita, y entonces deben ser presentados mediante una frecuencia específica durante un tiempo determinado o de lo contrario la integridad de la información se perderá.

De todos los contenidos multimedia, el más significativo por sus requisitos y características es el video. Un video consiste en una secuencia de imágenes que son visualizadas a una frecuencia preestablecida (play rate), que normalmente suele ser alrededor de 30 imágenes por segundo. Los contenidos multimedia tienen una naturaleza analógica y para que esta información pueda ser gestionada y almacenada en un ordenador debe ser digitalizada. Sin embargo, su digitalización genera un volumen de información demasiado grande para ser almacenada o transmitida eficientemente por la red.

Para reducir los requisitos de los videos, éstos se codifican guardando solo la información correspondiente a los píxeles o líneas de información consecutivas que son diferentes.

En cuanto al sonido, se eliminan los silencios y redundantes, para así aliviar la información gestionada. Las técnicas de codificación / compresión explotan las redundancias espaciales y temporales del video, las cuales pueden variar de una escena a otra.

3.2 Codificación de Video:

Los humanos vemos en forma continua la luz reflejada por los objetos, creándose los colores según la forma que éstos tengan para reflejarla. El video realiza algo parecido, pero captando secuencias discretas de imágenes, las cuales observándolas una tras la otra en forma rápida, producen la sensación de movimiento. Si proviene de una cámara grabadora (No webcam), ésta genera 30 cuadros por segundo NTSC. La tarjeta muestrea y obtiene las componentes de luminancia y crominancia para cada píxel. Las resoluciones son típicamente múltiplos o sub-múltiplos de 320x240 para NTSC.

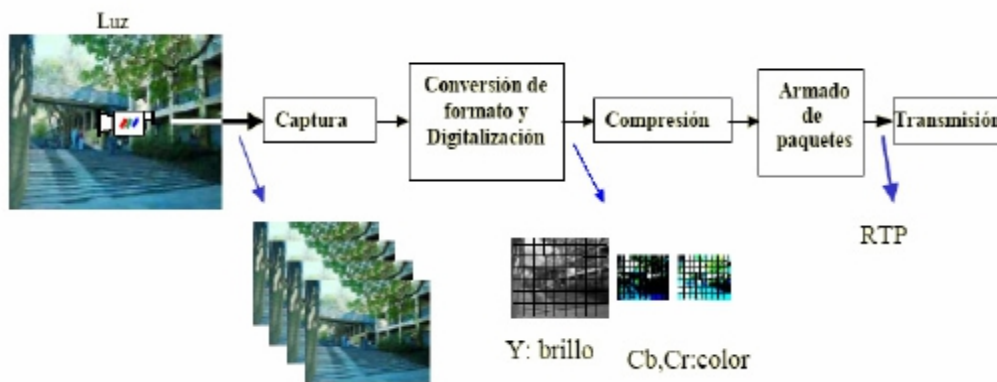


Figura 3.2.: Proceso de Codificación del Video.

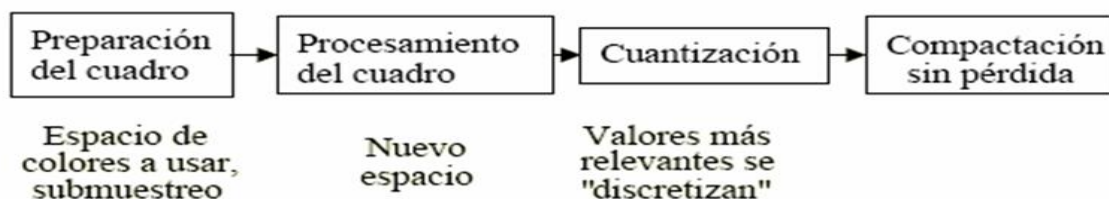


Figura 3.3.: Proceso de Compresión de cada Imagen del Video.

No necesitamos 30 frames por segundo por lo que 10 puede bastar. En muchos casos resoluciones de 320*240 basta. Cada cuadro o frame se subdivide en cuadrados de 8x8 píxeles.

Como notamos más los valores promedios que los detalles, éstos son codificados con mayor precisión, los detalles con menor precisión son eliminados. Se toman diferencias del cuadrado con respecto al cuadro anterior. Como la imagen se pudo correr, se busca el cuadro en un entorno (predicción del movimiento). Cada cuadro no cabe en un paquete, entonces se agrupan varios rectángulos autónomos.

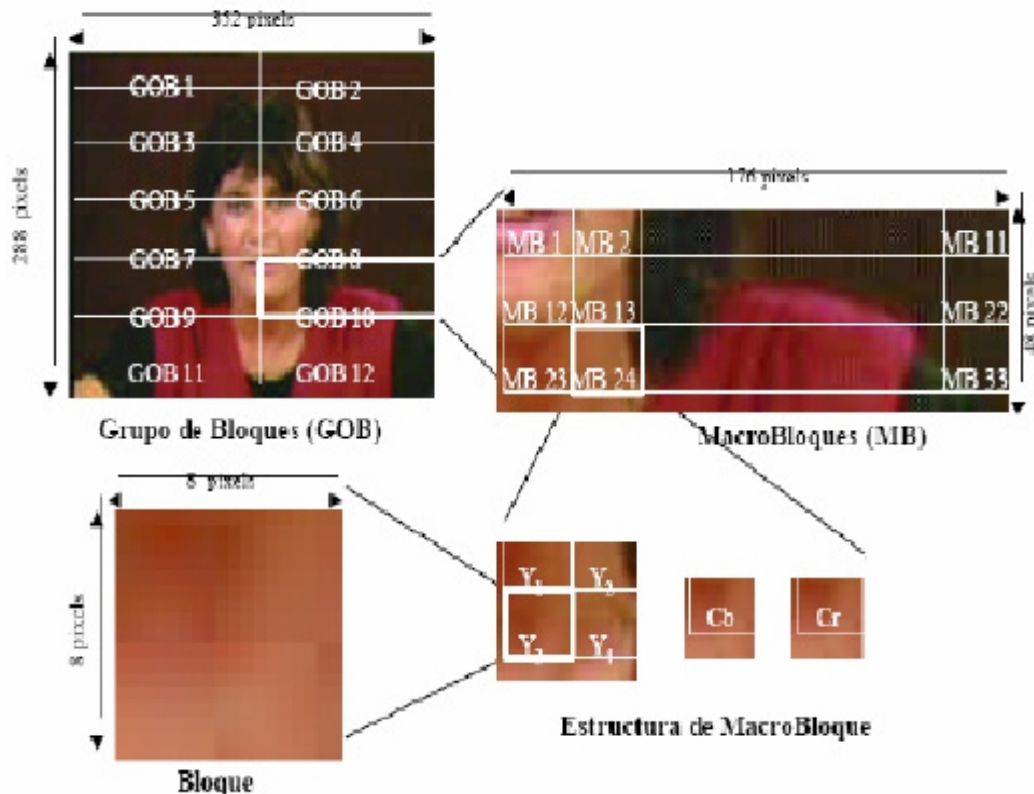


Figura 3.4.: Agrupación en Cuadros de una Imagen a Comprimir

Como hay pérdida de paquetes debemos enviar cada cierto tiempo cuadros comprimidos como imágenes

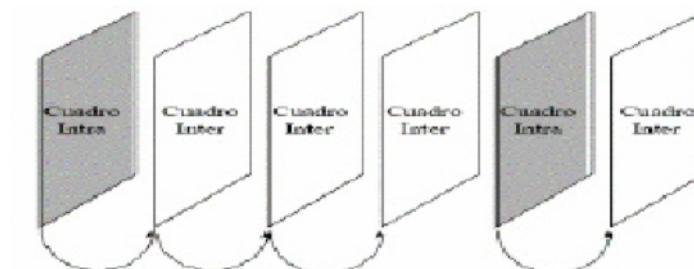


Figura 3.5.: Envío sucesivo de Paquetes Comprimidos de una Imagen.

El Receptor básicamente hace la operación inversa del proceso de codificación descrito anteriormente. El despliegue se hace en el monitor (RGB) por ello se ocupa mucha CPU en la descompresión y el cambio de formato de cada píxel. Hay alto movimiento de datos en los buses internos del computador que pueden frenar el sistema. El audio generalmente debe esperar al video para reproducción sincrónica

3.3 CODEC's utilizados por los Sistemas de Streaming:

El audio y el video se almacenan en las computadoras en forma de archivos. Debido a que la información digital audiovisual es un gran negocio, hoy se torna una "necesidad" almacenar cada minuto de audio y video en diferentes soportes; desde discos duros hasta CD-ROMs. Este almacenamiento debe ser "inteligente". No se trata nada más de copiar/capturar el material desde una unidad reproductora o emisora de señal (video casetera, lectora de DVD, lectora de CD, videocámara): también necesitamos "comprimir".

Un archivo de audio consiste en un "array" de números. Un array es una matriz de datos, todos del mismo tipo, posicionados desde cero por un entero. Cada uno de estos números representa el volumen y la frecuencia de sonido en un instante de tiempo. Puestos todos estos números juntos y ejecutados por el reproductor apropiado, generarán un flujo cambiante de frecuencias y volúmenes que serán entonces voz, música o efectos de sonido. Los archivos de video se comportan de manera similar, aunque utilizan los números para definir colores, brillo, contraste o coordenadas de cada parte de la cambiante serie de cuadros (frames) que componen una "película" o secuencia de imágenes.

La compresión digital del audio y del video se realiza de diferentes maneras. Al almacenar o leer un archivo de 'media' se aplican fórmulas matemáticas. Parte de estas fórmulas resuelven la compresión y descompresión de un archivo. Precisamente al software desarrollado en base a estas fórmulas matemáticas de compresión se le

denomina CODEC (por **CO**mpresor y **DE**Compresor)

Regularmente un CODEC es asociado a un formato de archivo en particular, pero un formato de archivo puede trabajar con más de un tipo de CODEC.

Mientras que algunos CODECs se basan en fórmulas matemáticas estándar, otros nuevos son creados en base a nuevos conceptos de compresión.

Es importante conocer qué CODEC tiene instalado en su PC para poder reproducir archivos de audio y video comprimidos. Es posible que en alguna ocasión Usted haya intentado leer un archivo y su sistema operativo haya respondido con un error o con un aviso de que el CODEC necesario para leer ese archivo está ausente. Más adelante veremos las características técnicas de los CODECs más populares. Los programadores experimentan constantemente con nuevas fórmulas y técnicas de compresión de audio y de video. En particular es interesante la compresión del audio, porque varias frecuencias de la música pueden ser eliminadas sin que el oyente lo tome en cuenta. Por ejemplo, si en una pieza de música predominan los tambores y tonos bajos se pueden recortar las frecuencias altas y nunca se sabrá lo que se está perdiendo; o si un pasaje en particular posee en su mayoría notas altas, Usted puede eliminar las frecuencias más bajas. Como regla general se eliminan aquellas frecuencias que el sistema auditivo humano no puede captar.

En el video el desarrollo de las fórmulas de compresión se enfoca en optimizar la reducción de la cantidad de colores. El ojo humano es más fácil de complacer que el oído: nuestra visión puede contentarse con una imagen de escasos 256 colores. Esta “dosis” de color es suficiente para delinear formas, contrastes, brillos y gamas con una definición que es aceptada de buen grado por nuestra interpretación visual. Más aún: por ejemplo, un silencio o distorsión en un sonido es detectado con facilidad, mientras que nuestra vista, ante una secuencia de imágenes incompleta o distorsionada en alguna de sus áreas, suele “auto completar” las piezas de información visual perdidas. Así es que analizando un sonido o una secuencia de imágenes cuidadosamente la fórmula de un CODEC puede ser muy eficiente al reducir el tamaño de un archivo.

Los CODECs se actualizan con el tiempo y utilizan una variedad de técnicas para

obtener la mejor compresión posible de un archivo. A modo de introducción se da a conocer algunos aspectos relevantes:

- Las tarjetas para captura de video aparecen desde hace más de una década, los primeros formatos fueron: Mov, Avi, Tga, Fli, entre otros. El formato AVI se convierte en el estándar pero al igual que los demás formatos requiere de grandes volúmenes de almacenamiento (utilizan codecs Intel-Indeo, Cinepak).
- MOV se convierte en el formato compatible para diversas plataformas (Mac, Windows, Linux).
- Más tarde aparece el formato MPEG original, usado para Video-CD que aportan una duración de 60 minutos de video en 650 MB.
- MPEG-2 se utiliza principalmente para los DVD's (archivos con extensión VOB), y alcanza el doble de resolución, así como un mayor nivel de compresión.
- Surge el MPEG-1 Layer 3 o MP3, que es lo mismo pero aplicado al sonido, comprimiéndolo de tal manera que un minuto de audio de alta calidad ocupa aproximadamente 1Mb.
- Microsoft crea el códec MPEG-4, que con la misma calidad de MPEG-2 comprime el video de forma sorprendente. Desde hardware (capturadoras, cámaras digitales) se puede alcanzar 1 hora de video en 64 MB. Existen 4 versiones de MPEG-4.
- En la comunidad Linux, dos programadores de Francia, derribaron la protección del códec MPEG-4 y lo mejoran apropiadamente para que cualquier tipo de usuario pueda disponer de él (licencia GNU). El nombre de esta remake del MPEG-4 se conoce como: Div-X.
- RealNetworks™ RealAudio® y RealVideo®, son los codecs utilizados por los programas entregados por RealNetworks, entre las extensiones que utilizan sus archivos son “.ra, .ram, .rm, .rmm”. Éstos son creados con una alta tasa de compresión y algoritmos especiales que reducen considerablemente el tamaño de los archivos de sonido y video. No tan famoso como el MP3 su capacidad de

streaming lo hace ideal para transmitirse en vivo a través de la red.

3.4 Tipos de servicios de Streaming:

Los sistemas de Streaming se pueden clasificar en función del tipo de servicio que ofrecen a los usuarios. La principal característica que distingue al servicio de Streaming, es la capacidad de interacción y elección de los usuarios a la hora de escoger “qué” contenido y “cuándo” lo quiere reproducir, aunque para la emisión de un canal de televisión en vivo, ésta afirmación no se aplica. Teniendo en cuenta éste parámetro los posibles servicios que puede ofrecer un sistema de Streaming son: “En directo” (live) similar a un canal de televisión, “bajo demanda” (on-demand) similar a un reproductor de vídeo y “casi bajo demanda”, que simula el funcionamiento de un servicio bajo demanda con flujos de vídeo en directo. El tipo de servicio ofrecido es un parámetro importante en el diseño, ya que a medida que se aumenta la interactividad del usuario también se incrementa la complejidad del sistema de Streaming y por lo tanto, el valor agregado del servicio ofrecido a los usuarios.

3.4.1 En directo (live):

Está orientado a la multidifusión, siendo éste tipo de servicio primordial para la emisión en directo de un canal de televisión. El servidor comienza a transmitir en un instante dado y los usuarios se conectan y ven la información que se está emitiendo en ese instante. En este tipo de servicio no existe interactividad, o sea que el usuario no puede adelantar o rebobinar, únicamente está permitido realizar pausas y cuando el usuario recupere la reproducción podrá ver la información que se está transmitiendo en ese instante. A continuación se ejemplifica el proceso de emisión en directo.

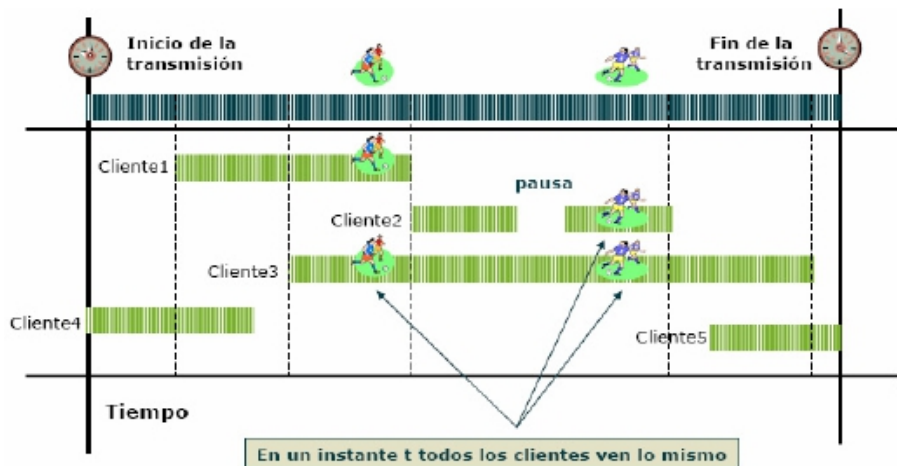


Figura 3.6: Proceso de una Emisión en Directo.

Las señales en vivo se pueden clasificar función de la fuente:

i. Según el origen de las señales de A/V:

La transmisión puede ser con información en vivo, o con información almacenada.

ii. Según el tipo de transmisión: La transmisión puede ser orientada a uno o a muchos usuarios, existiendo 3 tipos de transmisión:

- **Transmisión Unicast:** Consiste en la transmisión dedicada a cada usuario, o sea que se envían distintos flujos de streams a distintos usuarios por igual, dividiendo el ancho de banda entre ellos.
- **Transmisión Multicast:** Consiste en la transmisión por igual a un grupo usuarios, enviando el stream por la red a todos los usuarios que pertenezcan a aquel grupo y que deseen tomarla.
- **Transmisión Broadcast:** Consiste en la transmisión por igual a todos los usuarios, enviando el stream por la red a todos los usuarios que pertenezcan a aquel a la red y que deseen tomarla.

3.4.2 Bajo demanda(on-demand)

En tipo de servicio los usuarios solicitan el envío de información en el instante que lo deseen, siendo ésta información personalizada para cada usuario, siendo la base de la

televisión interactiva. Existen diversos tipos de interacciones:

- **Pausas:** Después de la pausa la reproducción se retoma en el punto donde se dejó.
- **Saltos hacia delante:** Es posible posicionarse en una zona más adelantada de la localización actual.
- **Saltos hacia atrás:** Es posible volver a visualizar zonas anteriores.

3.4.3 Casi bajo demanda:

Éste servicio simula el funcionamiento del vídeo bajo demanda mediante flujos de vídeo en directo, siempre con información almacenada. Cuando llega un cliente, se le incorpora al flujo que comienza (posiblemente tenga que esperar un pequeño intervalo de tiempo). Cuando realiza una interacción, se le incorpora al flujo que emite en la posición más cercana a la que solicita, cuando los flujos acaban la emisión, comienzan a transmitir otra vez por el principio (se realiza una emisión continua). Se utiliza para tratar de aprovechar las características de una emisión multicast.

3.5 Componentes de un sistema de Streaming.

Los sistemas de Streaming están compuestos por tres componentes básicos: el servidor, la red de transmisión y los usuarios del sistema. A continuación describiremos la funcionalidad de cada uno de estos componentes.

3.5.1 Servidor de Streaming:

El servidor de video almacena los contenidos que pueden ser solicitados por los usuarios. Es el encargado de gestionar el servicio a los clientes, garantizando una cierta calidad de servicio a lo largo del camino que tiene que seguir la información

desde el disco hasta los usuarios. Un servidor de Streaming está compuesto por tres subsistemas: El subsistema de control, el subsistema de almacenamiento y el subsistema de comunicación.

- i. **Subsistema de control:** El subsistema de control es el encargado de recibir las peticiones de los usuarios y ordenar las acciones que se tienen que llevar a cabo para poder atenderlas. Este módulo debe decidir si la nueva petición puede ser servida por el sistema sin que ello implique un deterioro de las peticiones activas. Estas decisiones son tomadas por la política de control de admisión en función de los recursos disponibles en el sistema y de los requisitos de la nueva petición. Otras funciones del módulo de control son la gestión de las estadísticas de utilización del sistema (contabilidad y facturación) y realización de tareas de optimización para incrementar la eficiencia del sistema.
- ii. **El subsistema de almacenamiento:** Este módulo es el responsable de almacenar y recuperar la información multimedia desde los dispositivos de almacenamiento. Las principales dificultades a la hora de conseguir este objetivo estriban en el volumen de información que se debe gestionar y que ésta debe ser entregada de acuerdo a las estrictas especificaciones de la calidad de servicio Streaming requeridas por las aplicaciones de video bajo demanda.
- iii. **El subsistema de entrega de comunicación:** Es el encargado de planificar la inyección de los contenidos multimedia en la red de transmisión. Este módulo se encarga de gestionar las distintas políticas de servicio que permiten optimizar los recursos de ancho de banda de la red y del servidor.

3.5.2 Red de comunicación:

Uno de los principales factores que más han influenciado en el crecimiento de las

aplicaciones multimedia es el crecimiento de la red de interconexión. Para permitir a los usuarios acceder a los contenidos multimedia, las redes deben satisfacer al menos dos requisitos: Disponer de mecanismos de transporte para enviar las peticiones y los datos y permitir que la información sea transmitida respetando niveles mínimos de rendimiento (calidad de servicio).

La red de comunicación de un sistema de Streaming se caracteriza por unos elevados requisitos de ancho de banda (capacidad de transferencia de grandes volúmenes de datos) y grandes velocidades de transmisión. En un sistema Streaming, podemos llegar a encontrar tres niveles de red diferentes: la red principal, la red troncal y las redes locales. Ahora bien, dependiendo de la arquitectura del sistema dos niveles (red principal y red troncal).

La red principal es aquella a la cual se conectan los servidores de Streaming y sirve punto de conexión de éstos con la red de distribución (red troncal) de los contenidos multimedia a los usuarios. La red troncal (o *backbone*) permite interconectar la red principal con cada una de las redes de distribución locales (en caso de que éstas existan) o bien directamente con los usuarios. Su objetivo es transportar, tan rápido como sea posible, la información generada por los servidores desde la red principal a los usuarios.

Las redes locales son las responsables de la conexión final de los usuarios al sistema de Streaming. Esta red requiere un ancho de banda inferior respecto a los otros niveles. El tráfico soportado por las redes de usuario tiene una naturaleza asimétrica, lo cual significa que se necesita un ancho de banda de entrada considerablemente mayor al tráfico de salida.

3.5.3 Clientes:

Los usuarios deben soportar la recepción y la visualización sin cortes de los contenidos multimedia, así como soportar los comandos VCR. La interface entre los

usuarios y el sistema de Streaming se realiza mediante el Player. Este módulo es el encargado de recibir los comandos del usuario y enviar la señal al servidor a través de la interface de red.

El Player almacena los contenidos recibidos desde el servidor en unos buffers locales, decodifica los contenidos recibidos en tiempo real y envía las imágenes obtenidas a la pantalla de visualización, con la temporización correcta.

En general los Streaming constan de 4 componentes principales: Interface de red, decodificador, buffer y hardware de sincronización.

- **Interface de red:** Permite al cliente recibir y enviar información desde ó hacia los servidores.
- **Decodificador:** Para reducir los requisitos de almacenamiento, ancho de banda de disco y ancho de banda de red, los contenidos multimedia suelen estar codificados. Así se necesita un decodificador en el lugar del cliente para decodificar el video antes de ser presentado al usuario.
- **Buffer:** Debido a los retrasos introducidos por la red, el tiempo de llegada de la información (video) no puede ser determinado con exactitud. Para conseguir una reproducción sin cortes, el servidor debe garantizar que el siguiente trozo del video que se va a visualizar, esté disponible antes que el usuario lo requiera. Para lograr este objetivo, el servidor envía datos al usuario en adelanto, de forma que se asegure un margen de tiempo que minimice los posibles retardos inesperados introducidos por la red de comunicación. Como el usuario no va a consumir inmediatamente estos datos, estos se tienen que almacenar temporalmente en un buffer hasta que sean requeridos.
- **Hardware de sincronización:** Los videos están compuestos por un stream de video y un stream de audio independientes. Para poder realizar un reproducción correcta, ambos tipos de información deben ser sincronizados entre si antes de que puedan ser reproducidos.

El desarrollo de los Player mantiene una continua evolución, no solo enfocada a

reducir su costo sino también a incrementar su potencia debido al rápido desarrollo tecnológico de la industria de ordenadores. Mientras las recientes generaciones de Player están bastante limitadas con respecto a la funcionalidad y a la capacidad, la actual tendencia intenta sobrepasar el mero rol de receptor y decodificador de video, convirtiéndolo en un verdadero centro de entretenimiento familiar, e incrementando su capacidad de almacenamiento y de procesamiento.

3.6 Requisitos de un Sistema de Streaming.

La funcionalidad requerida de un sistema de Streaming así como las características de la información gestionada por éstos, imposibilita la utilización de servidores genéricos. Por lo tanto, los servidores de Streaming deben ser diseñados teniendo en cuenta una serie de requisitos específicos del tipo de información gestionada. El servicio de una petición para un contenido multimedia requiere un elevado volumen de información, con requerimientos de tiempo real, mantenimiento de la calidad de servicio (QoS) y grandes anchos de banda de transferencia del sistema de almacenamiento y la red de comunicaciones.

El conjunto de todos estos requisitos complica el diseño e implementación de los sistemas de Streaming y limita considerablemente el número de usuarios que puede soportar un servidor de Streaming. A continuación se describen brevemente cada uno de estos requisitos.

- i. **Gran capacidad de almacenamiento:** Dada la naturaleza intensiva en almacenamiento de la información multimedia, los requisitos de almacenamiento globales de cientos de contenidos multimedia puede exceder fácilmente un requisito de disco de decenas de Terabytes. Por ejemplo, un video en formato de televisión de alta definición (HDTV) de dos horas de duración puede requerir hasta 18 Gigabytes. Por lo tanto, un sistema de Streaming compuesto de 200 videos puede requerir aproximadamente unos 3.6

Terabytes de almacenamiento.

ii. Servicio en tiempo real: Para garantizar la reproducción continua de los contenidos multimedia, no es suficiente con que el servidor de Streaming envíe los datos al usuario y éste los reciba correctamente; sino que esta recepción se debe producir dentro un intervalo de tiempo específico. Esto implica que todos los componentes del sistema deben tener un control del tiempo máximo permitido para poder realizar cada uno de las operaciones que intervienen en la entrega de información a los usuarios. Además, los distintos componentes que intervienen en el sistema se tienen que sincronizar entre sí para no violar estos requisitos de tiempo. Si esta sincronización no se lleva a cabo es imposible garantizar una calidad de servicio al usuario final. Es posible suavizar los requisitos en tiempo real de los sistemas de Streaming mediante la utilización de buffers intermedios tanto en el servidor como en el cliente y el envío en adelanto de un fragmento del contenido multimedia.

iii. Calidad de servicio (QoS): Un aspecto clave en cualquier servicio de vídeo es proporcionar una calidad de servicio (QoS) aceptable al usuario.

Debido a la naturaleza continua e independiente del tiempo de los contenidos de audio y video, su reproducción requiere un estricto control del momento y la secuencia de recepción de la información por parte del usuario. Esta calidad de servicio generalmente implica varios aspectos tales como: calidad de la imagen, frecuencia de pérdida de imágenes, sincronización audio y vídeo, entre otros. Algunos de estos parámetros no son fácilmente cuantificables porque dependen de la percepción subjetiva del observador. La calidad de servicio a nivel del usuario refleja cómo se suministra el flujo de vídeo original desde un servidor de vídeo remoto. Estos servicios requieren restricciones específicas en el flujo de información desde el servidor al cliente.

Por lo tanto, una cuestión importante en Streaming es como lograr una correspondencia entre la QoS específica requerida por el cliente con la especificación de una QoS para el servidor de vídeo y la red de transmisión.

Con el objetivo de conseguir unas prestaciones en el sistema que garanticen una QoS aceptable se requiere una fuerte coordinación entre todos los componentes del sistema, desde los servidores de ficheros a los dispositivos de visualización pasando por las componentes de red. No es suficiente un análisis individual de los componentes del sistema sino que se requiere un diseño unificado que tenga en cuenta todos los componentes. La QoS basada en el análisis de los componentes individuales tropieza con el problema de que los componentes no son independientes entre sí. La solución óptima para una componente no garantiza la mejor solución para todo el sistema y por lo tanto, se requiere un análisis integrado.

- iv. **Grandes anchos de banda:** Los contenidos multimedia requieren el procesamiento de un gran volumen de información de forma periódica y durante grandes periodos de tiempo. Este volumen de información exige grandes anchos de banda en la red de transmisión. Los requisitos de ancho de banda no se circunscriben exclusivamente a la red de comunicaciones entre el servidor de Streaming y los usuarios finales, sino que también involucran al sistema de almacenamiento. Esto implica la utilización de sistemas de almacenamiento complejos basados en sistemas de almacenamiento jerárquicos o bien la utilización de un conjunto de discos en configuración RAID. Es importante hacer notar que si no se tiene en cuenta este parámetro en el diseño del sistema de Streaming, un incremento en el número de peticiones a gestionar por el sistema, puede aumentar los requisitos de ancho de banda hasta llegar a saturar el sistema de Streaming.

3.7 Formatos de Archivos de Streaming.

Vamos a ver cuáles son las características de diferentes formatos de "meta archivos" y de archivos de audio y video, la lista de formatos no es exhaustiva.

Solo trataremos los formatos de uso más populares en los servicios de Streaming y en

aplicaciones multimedia.

3.7.1 Metafiles – Meta archivos:

META es una palabra griega que significa "próximo" o "cercano". Un meta-archivo o meta-file es un tipo especial de archivo que describe o brinda más información acerca de otro archivo. Los METAARCHIVOS o METAFILES son archivos de texto que utilizan XML (Extended Markup Language) para definir tipos de información específicos. Los metafiles proveen varias maneras de hacer más eficiente el funcionamiento de un Player. Algunas de las extensiones más populares:

i. **Windows Media:**

ASX: Contienen información acerca de archivos ASF. La extensión ASF fue usada para archivos de audio y video por versiones anteriores a la actual de Windows Media.

WAX: Contienen información acerca de archivos de audio. Cuando Usted graba una lista de ejecución de archivos de audio (playlist) Windows Media Player utiliza este formato.

WVX: Contienen información acerca de archivos de video. Cuando Usted graba una lista de ejecución de archivos de video (playlist) Windows Media Player utiliza este formato.

ii. **Real:**

RAM/RPM: Contienen información acerca de archivos RM que pueden ser tanto de audio como de video o ambos.

El uso más corriente de los metafiles es el de archivo para lectura y escritura de "playlists" en el disco duro. Un archivo PLAYLIST contiene una lista de directivas que el reproductor de audio/video debe ejecutar en el orden y forma que este archivo (playlist) le indique. Las acciones o aplicaciones más comunes de un playlist son:

- Vincular contenidos multimedia para enriquecer una presentación.
- Insertar publicidad (gráficos, audio, video) en la pantalla principal del reproductor, entre los archivos de contenido principales (como si se tratara de cortes comerciales de televisión); o insertar publicidad en áreas específicas de la pantalla del reproductor o página Web -si el archivo estuviera incrustado en el documento HTML- *mientras* se ejecuta el archivo de audio / video (los comerciales pueden contener enlaces hacia páginas Web o direcciones de email)
- Titular o subtítular archivos de video: útil para la traducción del contenido audiovisual (un meta-file puede incluso permitirle al usuario seleccionar el idioma), para personas con discapacidad auditiva o para resaltar secciones en un archivo de audio o destacar escenas en un archivo de video.
- Enviar notificaciones o comunicados en streaming media: por ejemplo, en un evento que será transmitido en vivo, un meta-file puede especificar cuándo y en que canal tomará lugar la transmisión; así el reproductor puede comenzar a recibirla en el momento debido.

Un playlist puede ser enviado por email o estar alojado en un servidor para descargarlo a pedido.

3.7.2 Archivos de Audio:

i. Windows Media:

- **ASF:** Este formato fue usado en versiones previas de Windows Media.
- **WMA:** Ambos formatos poseen información idéntica excepto que el segundo formato es solo de audio, mientras que el primero puede contener información de audio y de video.

ii. Real:

- **RM**
- **RA:** Este formato fue usado en versiones previas de Real.

Ambos formatos poseen información idéntica excepto que el segundo formato es solo de audio, mientras que el primero puede contener información de audio y de video.

iii. **Microsoft Windows:**

- **WAV:** Formato nativo de ese sistema operativo.

iv. **Apple Macintosh:**

- **AIFF:** El formato de intercambio de audio o "Audio Interchange File Format" es uno de los más populares de Apple Macintosh. Este formato utiliza las extensiones .AIF, .AIFC y .AIFF

v. **Unix:**

- **AU:** Este formato utiliza la extensión .AU. El formato .SND es muy similar. Ambos son populares en UNIX y computadores tipo Unix.

vi. **Audio MPEG:**

- **MP3:** Los archivos MP3 son creados de acuerdo con el estándar del Grupo de Expertos en Imágenes en Movimiento o "Moving Picture Experts Group" (MPEG), conocido como LAYER 3. Este formato utiliza las extensiones .MP3 o .M3U.

3.7.3 Archivos de Video:

i. **Windows Media:**

- **ASF:** Este formato fue usado en versiones previas de Windows Media.
- **WMV:** Ambos formatos poseen información idéntica excepto que el segundo Formato es solo de video, mientras que el primero puede contener información de audio y de video.

ii. **Real:**

- **RM**

iii. **Microsoft Windows:**

- **AVI:** Formato nativo de ese sistema operativo. Contiene información de audio y de video.

iv. **Video MPEG:**

El video creado bajo las especificaciones del Grupo de Expertos en Imágenes en Movimiento o Moving Picture Experts Group (MPEG) puede tener las siguientes extensiones de archivo: **MPG, MPEG, MPE, M1V, MP2, MPA, MPE**

v. **Apple Macintosh:**

- **MOV:** Es uno de los más populares de Apple Macintosh. Este formato utiliza también las extensiones **.MOV, .QT.**

vi. **Indeo:**

- **IVF:** Indeo ha sido adquirido recientemente por Ligos Technology.

4. POLÍTICAS DE GESTIÓN DE LOS SERVIDORES DE STREAMING:

Para que un servidor pueda efectuar sus funciones, éste debe realizar una secuencia de tareas periódicas cada una de ellas sujetas a una estricta temporización de forma que se garantice un flujo de información continuo a lo largo de todo el camino de servicio que debe seguir la petición. Este camino de servicio se inicia en el subsistema de almacenamiento, pasando por el subsistema de inyección en la red que se encarga de enviar los contenidos multimedia a los clientes a través de la red de comunicación.

A continuación se muestran los distintos módulos que componen un servidor de Streaming: el módulo de control de admisión, los planificadores de disco y de red y el gestor del almacenamiento. Las funciones realizadas por cada uno de estos módulos y algunas de sus políticas más significativas se comentan en los siguientes apartados.

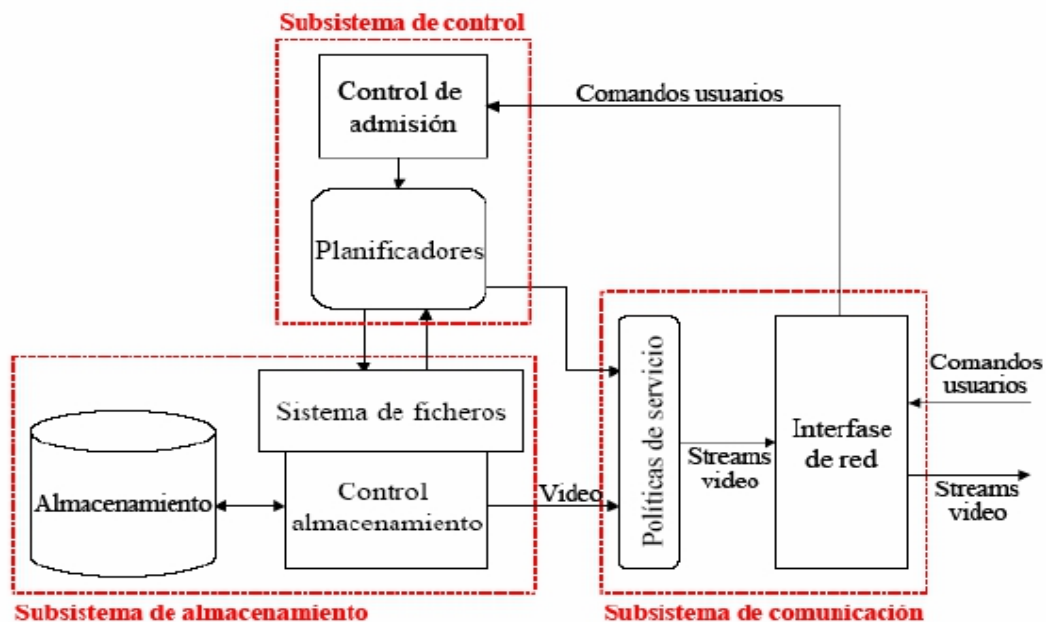


Figura 4.1: Módulos que Componen un Sistema de Streaming.

4.1 Control de Recursos:

Para garantizar que los nuevos clientes disponen de un servicio continuo y asegurar que la calidad de servicio asignada a las conexiones existentes se mantiene, el servidor debe garantizar que dispone de suficientes recursos (ancho de banda de E/S de disco, buffers de memoria, ancho de banda de procesamiento y ancho de banda de red) antes de admitir una nueva petición. La toma de la decisión de si una petición se puede o no servir con los recursos actualmente disponibles, recae bajo la responsabilidad del módulo de control de admisión del servidor.

Antes de admitir una nueva petición, un conjunto de parámetros de QoS son transmitidos por el usuario para que sean comprobados por el servidor. Si esta calidad de servicio pedida no puede ser soportada, la petición será denegada ó alternativamente, se establecerá un proceso de negociación entre el servidor y el usuario para reducir los requisitos solicitados.

El algoritmo de control de admisión debe disponer de conocimientos sobre la capacidad del subsistema de almacenamiento y la política de servicio activa de forma que se pueda evaluar adecuadamente el volumen de recursos requeridos para atender la nueva petición (en el caso de que se puedan utilizar políticas de multicast que permitan reducir el consumo de recursos).

4.2 Planificadores:

En un modelo genérico, el servidor de video se compone de un planificador de disco y de un planificador de red. El planificador de disco determina cómo y cuándo se tienen que transferir desde el subsistema de almacenamiento hasta los buffers intermedios de memoria. El planificador de red determina cómo y cuándo la información se transfiere desde los buffers de memoria a la red, para su transmisión a los clientes.

i. Planificador de disco:

Para una eficiente utilización del ancho de banda de disco, el planificador de disco se suele organizar en ciclos. En su forma más sencilla, el esquema basado en ciclos consiste en que en cada ciclo se planifican y se leen los datos que necesita transmitir el planificador de red en el siguiente ciclo. El principal motivo para esta organización en ciclos es que permite desligar el orden de transmisión de los datos a los usuarios del orden de lectura desde los dispositivos de almacenamiento. Al poder leer los bloques del disco en cualquier orden se puede minimizar los tiempos de búsqueda (*seek*) en el disco. El orden de lectura de cada uno de los bloques del disco en cada uno de los ciclos dependerá de la política concreta que se aplique. Se pueden utilizar políticas genéricas como el SCAN, EDF o SCAN-EDF, o bien políticas específicas orientadas a sistemas de tiempo real como GSS (Grouped Sweeping Scheme).

ii. Planificador de red :

Los dos principales objetivos de las políticas (presuponiendo la correcta recepción de los usuarios) son simplificar la gestión del sistema de Streaming (suavizando los requisitos de tiempo real o la variabilidad de la frecuencia de compresión de los contenidos Multimedia) y reducir los recursos requeridos para servir las peticiones.

Una de las técnicas más conocidas que permite reducir los requisitos de ancho de banda, es el *smoothing* (suavizado). Esta técnica utiliza el buffer del cliente para enviarle datos (trozo del video) por adelantado, minimizando la posibilidad de que el usuario perciba un retardo. El objetivo de estos datos enviados por adelantado, es permitir que el planificador de red pueda minimizar la fluctuación en los requisitos de ancho de banda de red de los contenidos multimedia.

El *prefetching* es otra técnica que intenta enviar datos por adelantado al servidor, pero en este caso, con el objetivo de reducir los requisitos de tiempo real del sistema y

optimizar los recursos del sistema, aprovechando los periodos de tiempo en los cuales el sistema está inutilizado.

Para reducir los requisitos de ancho de banda del sistema, el planificador suele implementar distintas políticas de servicio, algunas de las cuales, comentamos en el siguiente apartado.

4.3 Políticas de servicio:

Las políticas de servicio son las encargadas de decidir cómo se deben gestionar las peticiones de los usuarios y el tipo de servicio que finalmente ofrece el sistema.

Existen tres tipos de categorías de políticas de servicio, en función del tipo de comunicación utilizada: Unicast (1 a 1), Broadcast (1 a todos) y multicast (1 a n):

- i. **Unicast:** Es la política de servicio más sencilla, ya que se basa en enviar un flujo de información independiente (mediante una transmisión Unicast) para cada una de las peticiones. La principal ventaja de esta técnica estriba en que soporta Streaming verdadero y los comandos del usuario. En el lado negativo, podemos destacar su poca eficiencia con respecto a la utilización de los recursos del sistema y a los grandes anchos de banda requeridos para servir a un número elevado de usuarios.
- ii. **Broadcast:** Esta política intenta maximizar la eficiencia de los recursos del sistema a costa de la interactividad de los usuarios. Se utiliza principalmente para ofrecer servicios Streaming de bajo costo. Las políticas de Broadcast se basan en las comunicaciones con el mismo nombre, en las cuales se permite enviar un mismo flujo de datos a todos los usuarios de una red, de forma indiscriminada. Los receptores deben decidir si la información les interesa o no. Si no es así, entonces sencillamente descartan la información recibida. Lo cual implica que los streams utilicen ancho de banda de la red, tanto si van a ser utilizados por usuarios como si ningún usuario los utiliza. Esta característica condiciona la utilización de estas políticas ya que requieren que la información que se está transmitiendo tenga una alta frecuencia de acceso para obtener un

rendimiento acorde con el ancho de banda utilizado. Ésta es la principal razón por la cual esta técnica solo se emplea con videos cuya popularidad es muy alta. Existen diversas técnicas de Broadcast, las cuales se diferencian principalmente en la forma en que se fracciona y se transmite el contenido multimedia. Los principales parámetros con los que juegan estas técnicas son el número de streams utilizados en el Broadcast de la película y la frecuencia de transmisión de cada uno de los trozos. La técnica de *Broadcast Pyramid* se basa en la existencia de buffers en los clientes, de forma que puedan recibir y guardar algunos segmentos del video mientras se visualizan los primeros.

Se transmiten segmentos cada vez más largos de cada video por canales diferentes. De esta forma, se divide el ancho de banda del servidor en N canales lógicos y la película en N segmentos con un tamaño que se incrementa geométricamente.

Debido a que el primer segmento es el más pequeño su frecuencia de transmisión será mayor, lo cual asegura un tiempo de espera considerablemente más reducido.

El esquema de broadcasting Skyscraper intenta reducir los requerimientos de ancho de banda y el tamaño del buffer del cliente de la técnica anterior. Se basa en dividir el ancho de banda disponible en B canales lógicos los cuales se distribuyen equitativamente entre los M videos. Para transmitir un video sobre sus K canales dedicados, éste es partido en K fragmentos, los tamaños de los cuales siguen una determinada serie. Además de las técnicas descritas, existen otras alternativas como

Broadcast pagoda ó Broadcast armónico. Las últimas tendencias intentan utilizar conjuntamente técnicas de broadcast y Unicast para proporcionar servicios de Streaming verdadero.

- iii. **Multicast:** Mediante las técnicas de multicast el flujo de información solo se envía (de forma discriminada) a un grupo de usuarios que han pedido los mismos contenidos. De esta forma, un stream de multicast siempre tiene por lo

menos un destinatario y nunca se malgasta ancho de banda. Esta es la principal razón por la cual estas técnicas suelen tener rendimiento más eficiente que las técnicas de broadcast. A continuación vamos a describir algunas de las políticas multicast más significativas.

La política de *batching* se basa en retrasar las respuestas (transmisión de los contenidos) a los usuarios de forma que varias peticiones a un mismo video se puedan servir utilizando un único flujo de información. La principal ventaja de esta técnica estriba en que se permite un considerable ahorro de recursos del sistema Streaming y que no requiere que el Player del usuario disponga de unas características específicas. En su contra está que no puede ofrecer un servicio de Streaming verdadero y que puede causar la cancelación de la petición por parte del usuario si éste no está dispuesto a esperar durante más tiempo por el servicio solicitado.

La política de *patching* es una de las primeras políticas de multicast que soporta servicios de Streaming verdadero. Esta política se basa en aprovechar los flujos de información que se están transmitiendo a otros usuarios para reducir los requisitos de las nuevas peticiones. Cuando el servidor recibe una nueva petición se comprueba si existe algún stream que esté transmitiendo el mismo contenido y que esté dentro de la ventana de tiempo del buffer del nuevo usuario (el minuto X de la información que se está transmitiendo no tiene que ser mayor que la capacidad del buffer). Si esta condición se cumple, el nuevo usuario se añade al canal de transmisión activo (con lo cual se asegura la recepción de todo el resto del video, a partir del minuto $X-120$, mediante un multicast) y se crea un canal específico para el trozo de video que falta (canal de patch ó parche) de forma que se pueda empezar inmediatamente la reproducción del video.

Esta política tiene una serie de ventajas: el mínimo tiempo de respuesta a los clientes, que permite expandir el canal de multicast para servir a nuevas peticiones y que la mayoría de los canales de patch son de corta duración. En

contraposición, esta técnica requiere que el Player del usuario disponga de un buffer para guardar alrededor de 5 minutos de video (tamaño normalmente utilizado mediante esta técnica) y además debe poder soportar la recepción simultánea de al menos dos canales de transmisión de entrada (canal de patch + canal multicast).

Existen otras técnicas multicast que permiten crear una estructura jerárquica de multicast (política de *merging*); o que logran la fusión de las peticiones de los usuarios en un canal multicast incrementando el volumen de información enviada al usuario (*piggybacking*) ó bien incrementado el ratio de compresión de los contenidos enviados (*skimming*). Con respecto a las políticas de servicio multicast, otra importante línea de investigación actual analiza la influencia de los comandos de los Player sobre la efectividad de esta política.

4.4 . Sistema de almacenamiento:

Debido al volumen de información gestionado en un sistema de Streaming, el sistema de almacenamiento puede resultar muy costoso si exclusivamente se utilizan discos magnéticos. Para construir un sistema de Streaming con una buena relación costo / prestaciones resulta lógica la utilización de sistemas de almacenamiento híbridos que combinan dos o más tipos de dispositivos de almacenamiento. Este esquema configura un sistema jerárquico de almacenamiento constituido por distintos niveles: memoria (formado por los buffers internos del servidor), discos magnéticos (para almacenar los videos con una popularidad media-alta) y los discos ópticos (para aquellos contenidos con la frecuencia de acceso más bajas). El número de peticiones concurrentes que puede ser gestionado por un único disco está limitado por su rendimiento, es decir por el ancho de banda del disco requerido para servir un contenido multimedia. Una aproximación que se puede utilizar para solucionar esta limitación consiste en mantener múltiples copias de un contenido en diferentes discos, sin embargo esta solución resulta cara. Una mejor aproximación consiste en fraccionar los ficheros

multimedia entre múltiples discos utilizando las técnicas de *striping*, *interleaving* ó una combinación de ambas. La técnica de *striping* utiliza la tecnología RAID (Rendundant Array of Inexpensive Disks) que permite realizar accesos en paralelo a un array de discos. Mediante el *interleaving*, los bloques de un fichero multimedia son almacenados de forma intercalada a lo largo de un conjunto de discos.

5. ARQUITECTURAS UTILIZADAS EN LOS SISTEMAS DE STREAMING:

En los apartados anteriores hemos descrito los principales componentes y políticas que integran un sistema de video bajo demanda. Sin embargo, a la hora de implementar un sistema de estas características se pueden adoptar diversas arquitecturas. En este apartado describiremos la organización y características de las principales arquitecturas utilizadas para el diseño e implementación de los sistemas Streaming.

5.1 Arquitecturas centralizadas:

Estos sistemas se basan en la conexión de todas las redes de usuarios del sistema a una red principal a la cual se conecta un servidor ó un conjunto de servidores.

Las principales características que definen las configuraciones centralizadas son la gestión centralizada de todas las peticiones de los usuarios y la utilización de una red principal que es compartida por todos los flujos de información del sistema. Existen dos categorías de sistemas centralizados en función del número de servidores utilizados: arquitecturas con un único servidor y arquitecturas basadas en múltiples nodos de servicio (o arquitectura de servidor distribuida).

En la primera configuración con un único servidor, la gestión de los clientes se basa en un único nodo de servicio que centraliza la atención de todas las peticiones. En estas arquitecturas, el servidor de video utilizado puede variar desde un PC estándar (para sistemas de pequeña escala) hasta supercomputadoras con cientos de procesadores (sistemas de gran escala). Sin embargo esta aproximación, en general, tiene diversas limitaciones con respecto a la escalabilidad, tolerancia a fallos y disponibilidad del

servicio. Para evitar los inconvenientes asociados con la utilización de un único servidor en la literatura se ha propuesto la utilización de múltiples nodos de servicio, de forma que se logre un servidor escalable (mediante la inclusión de nuevos nodos de servicio), tolerante a fallos (ya no se depende de un único punto de fallo) y que puede alcanzar una mayor capacidad de servicio. Dentro de las arquitecturas centralizadas basadas en múltiples nodos de servicio podemos encontrar dos configuraciones diferentes en función de cómo se organizan los distintos nodos: servidores paralelos (o array de servidores) o formando un cluster.

En general los servidores centralizados obtienen mejores rendimientos con respecto a la probabilidad de bloqueo de las peticiones siempre que ambas configuraciones dispongan del mismo ancho de banda de E/S. Sin embargo los servidores distribuidos tienen una mejor escalabilidad, una alta disponibilidad, un mejor costo y pueden alcanzar el mismo rendimiento que los servidores centralizados incrementando su capacidad de almacenamiento o de E/S.

El principal problema que sufren las arquitecturas centralizadas es el cuello de botella que representa la red principal. La escalabilidad futura del sistema queda limitada por el ancho disponible en esta red.

i. Servidores paralelos ó array de servidores:

Esta arquitectura consiste en un array de servidores, que trabajan de forma similar a un array de discos. Los distintos nodos de servicio no almacenan videos completos, sino que los videos son divididos en trozos y éstos son distribuidos entre los diferentes nodos para lograr una distribución de la carga más homogénea entre todos los servidores. A continuación se muestra la configuración de una arquitectura de servidores paralelos compuesta por 5 servidores, todos ellos conectados con los usuarios a través de una red de interconexión. Cada uno de los servidores almacenan un subconjunto de segmentos (v_1, v_2, \dots, v_n) de los videos del catálogo el sistema.

Cuando se produce una petición cada nodo de servicio es el responsable de transmitir al usuario los fragmentos de video solicitado que se encuentran en sus discos. El

cliente tiene la responsabilidad tanto de fraccionar su petición en las distintas solicitudes a cada uno de los nodos de servicio, como de posteriormente recombinar y sincronizar los distintos flujos de información recibidos, para poder reproducir el contenido multimedia. Esta arquitectura permite escalar la capacidad del sistema añadiendo nuevos nodos, aunque se requiere realizar una nueva redistribución de los videos que tenga en cuenta los nuevos servidores añadidos.

Otras ventajas asociadas con esta configuración son que permite un balanceo automático de la carga del sistema y que aumenta la tolerancia a fallos respecto a las arquitecturas basadas en un único nodo de servicio. La principal desventaja de estas aproximaciones es que incrementan los requisitos de los Player de los usuarios y complica considerablemente su diseño. Para evitar la utilización de Players demasiados complejos se ha propuesto la utilización de proxies entre los servidores y los clientes. En estos sistemas, el término proxy hace referencia al módulo del sistema encargado de re-secuenciar y fusionar los datos procedentes de los distintos servidores en un flujo de información coherente para entregárselo al usuario final. El proxy puede también utilizar información redundante para enmascarar posibles fallos en los servidores. Además, al no existir un nexo común en la gestión de las peticiones de los usuarios, la utilización de políticas de compartición de recursos se complica considerablemente.

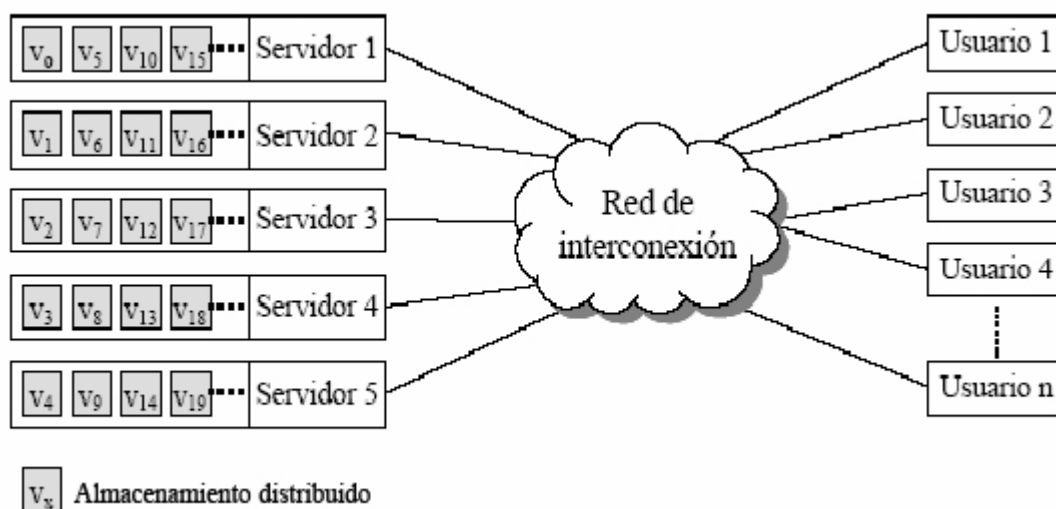


Figura 5.1: Esquema Servidores paralelos o array de servidores.

ii. Cluster de servidores:

Una arquitectura basada en cluster consiste en un grupo de nodos conectados entre sí por una red de interconexión. Cada nodo dispone de un disco local conectado a él. Los nodos del cluster se pueden dividir en tres categorías: nodos de transmisión y nodos de almacenamiento y un nodo de control. El nodo control admite las peticiones de la red externa basándose en una estrategia de control de admisión predefinida o dinámica. Los nodos de almacenamiento, guardan los contenidos de forma similar a los servidores paralelos (i.e. cada objeto multimedia es dividido en bloques y distribuido entre todos los discos del sistema), proporcionándolos cuando son requeridos a los nodos de transmisión. Los nodos de transmisión son los encargados de unir los distintos bloques correspondientes a un video, antes de su transmisión al usuario en forma de un único flujo de información.

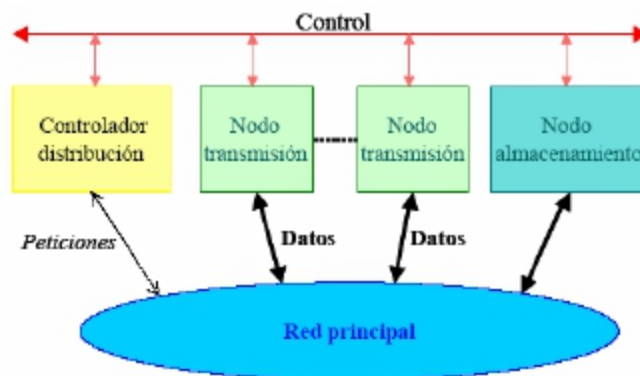


Figura 5.2: Esquema Cluster de Servidores.

5.2 Arquitecturas de servidores independientes:

Una de las soluciones que se ha propuesto para incrementar la escalabilidad de los sistemas de Streaming, es la conexión de los usuarios mediante servidores independientes. En estos sistemas, tal y como se muestra a continuación, los usuarios

están agrupados en segmentos de red cuyo tráfico es independiente entre sí, denominados redes locales, de forma que el ancho de banda del sistema pueda llegar a ser el ancho de banda acumulado de cada una de las redes individuales. Sin embargo, para incrementar el ancho de banda de estos sistemas no es suficiente con únicamente agrupar los usuarios en redes independientes ya que si todos acaban accediendo al mismo servidor y a su red, éstos se convierten en el cuello de botella del sistema, con la consiguiente saturación de todo el sistema. La clave para que estos sistemas con redes independientes funcionen y obtengan un mejor rendimiento, estriba en que las peticiones se puedan servir localmente sin la necesidad de acceder a un servidor centralizado. Este objetivo se puede lograr colocando servidores de Streaming cerca de las redes locales de los usuarios y replicando todos los contenidos, de forma que éstos no tengan que acceder al servidor central, creando un sistema de servidores independientes o autónomos.

Las principales ventajas de esta arquitectura es que permite una escalabilidad ilimitada mediante la inclusión de nuevos servidores a los cuales se conectarán los nuevos usuarios y que no requiere servidores muy complejos. Por el contrario, los sistemas de Streaming basados en servidores independientes tienen unos elevados costos asociados con el subsistema de almacenamiento debido a que todos los servidores deben replicar los contenidos del catálogo del sistema. Una alternativa para reducir estos costos consiste en interconectar los servidores entre si y permitir que los servidores sólo almacenen un subconjunto de los videos del sistema, redirigiendo las peticiones que no se puedan servir localmente hacia los otros servidores del sistema. De todas formas, para no saturar la red de interconexión entre los servidores, estos aún necesitan almacenar un porcentaje considerable de los contenidos del sistema.

5.3 Arquitecturas basadas en servidores-proxy:

La arquitectura de servidores independientes implica un elevado costo, y por lo tanto, algunas propuestas han optado por reducir el tamaño de los servidores locales, de forma que no almacenen una copia completa de las películas del sistema, sino

únicamente los contenidos más populares. Estos servidores locales se denominan servidores-proxy, al igual que sus homólogos de Internet y se comportan como una caché del catálogo de contenidos almacenado en un servidor principal, el cual contiene todos los videos disponibles en el sistema. Los sistemas basados en servidores-proxy de un nivel, identifican a una arquitectura en la cual los servidores-proxy no están interconectados entre sí. Esta arquitectura surge como un compromiso entre las arquitecturas centralizadas (no escalables pero con menores almacenamiento) y las arquitecturas de servidores independientes (escalables, pero con elevados costos de almacenamiento). Los servidores-proxy son los encargados de gestionar inicialmente todas las peticiones generadas por los usuarios conectados a sus redes (redes locales), en el caso que la petición no pueda ser atendida localmente debido a que el contenido requerido no se encuentra en la caché, entonces se redirige la petición hacia el servidor principal. Existen diversas políticas para gestionar el contenido de los servidores-proxy en función de si almacenan los contenidos completos o solo un fragmento: prefix-caching basado en almacenar en la caché el fragmento inicial (prefijo) de los contenidos de video más populares y segment-caching que archivan en la caché los fragmentos del video más populares.

Existen dos configuraciones básicas que se pueden utilizar a la hora de diseñar un sistema Streaming basado en servidores-proxy. Ambos difieren en la arquitectura utilizada para el servidor principal a la cual se conectan los distintos servidores-proxy. Tenemos la arquitectura de servidores-proxy basados en un servidor centralizado y las arquitecturas de servidores-proxy basadas en un servidor paralelo o jerárquico.

i. Basados en un servidor principal centralizado:

La topología general de un sistema basado en servidores-proxy se compone de un servidor principal al cual se conectan directamente y a través de una red principal, un conjunto de redes locales con su proxy. Debido a que solo hay un nivel de servidores-proxy en la arquitectura, este sistema se suele denominar, sistema basados en

servidores- proxy de un nivel (en contraposición a otras arquitecturas que pueden utilizar diferentes niveles jerárquicos de servidores-proxy dentro del sistema).

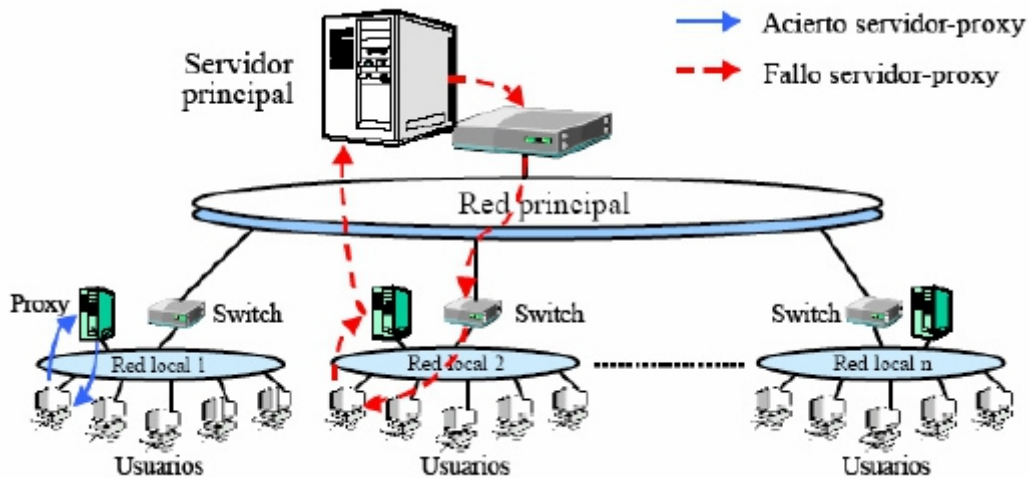


Figura 5.3: Esquema Arquitecturas basados en un servidor principal centralizado

Como se muestra en la figura también se pueden ver los dos tipos de peticiones que tienen estos sistemas: peticiones servidas localmente y las peticiones atendidas por el servidor principal. Las peticiones locales (trazo continuo) son aquellas que se pueden servir desde los contenidos almacenados en la caché de los servidores-proxy. Cuando el servidor-proxy no dispone del contenido requerido por la petición, ésta se redirige hacia el servidor principal que se encargará de su servicio. Este tipo de peticiones requieren el doble de ancho de banda de red para ser atendidas. El principal problema que se encuentran estos sistemas es la escalabilidad limitada derivada de la dependencia de unos componentes centralizados como son el servidor y la red principal. La capacidad de crecimiento del sistema dependerá en última instancia de la capacidad de estos componentes centralizados. De todas formas, siempre disponen de un mayor margen de maniobra comparado con los sistemas centralizados.

ii. **Basados en un servidor principal paralelo / jerárquico:**

Esta aproximación trata de solventar los problemas de escalabilidad del servidor principal centralizado en la arquitectura de servidores-proxy de un nivel. En esta

arquitectura, de la cual la figura que se muestra a continuación muestra una posible configuración, el servidor principal está diseñado basándose en una red jerárquica o en árbol, con servidores de Streaming en los nodos y enlaces de red en las ramas de la jerarquía. Los nodos de servicio situados en las hojas de la jerarquía son los puntos de acceso para el sistema. Todos los nodos del sistema solo almacenan un subconjunto de los contenidos del sistema.

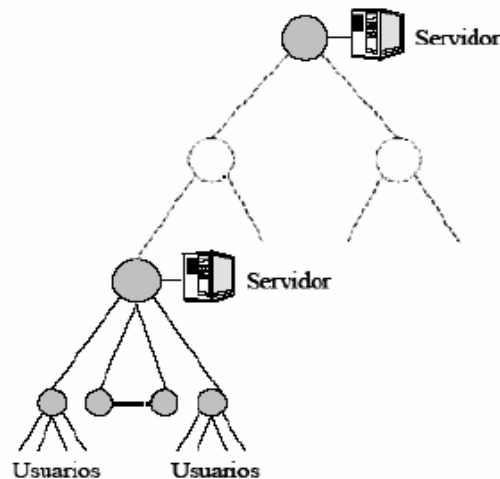


Figura 5.4: Esquema Arquitecturas basados en un servidor principal paralelo / jerárquico.

Cuando una petición para un contenido llega a un nodo hoja, si el contenido está disponible en su almacenamiento local, el servidor atiende el mismo al cliente. En caso contrario, la petición se reenvía hacia los niveles superiores de la jerarquía para que sea atendida por otro nodo de servicio de la arquitectura que disponga del contenido requerido. El rendimiento de esta arquitectura es similar a las basadas en servidores-proxy de un nivel conectadas a un servidor centralizado, pero reduce la probabilidad de saturación del servidor principal e incrementa la capacidad de servicio del sistema.

5.4 Arquitecturas distribuidas a nivel de los usuarios:

Las últimas tendencias a la hora de diseñar sistemas de Streaming se orientan hacia la adopción de arquitecturas distribuidas, en las cuales la gestión de las peticiones, así como los contenidos multimedia se distribuyen entre todos los componentes del

sistema.

En estos sistemas, los distintos nodos de servicio tienen que colaborar entre si para poder atender a los usuarios. Existen diferentes categorías de sistemas distribuidos en función de si existen o no un nodo maestro encargado de centralizar la gestión del sistema y mantener una copia completa de los contenidos del sistema (arquitecturas de servidores-proxy de un nivel, por ejemplo). Una de las primeras propuestas en este sentido es la política de servicio de *Chiang* (encadenamiento). Esta política utiliza el contenido de los buffers internos de los Players de los usuarios para a su vez servir peticiones de otros usuarios hacia el mismo contenido. De esta forma, se crea una cadena de servicio (cada eslabón de la cadena consiste de un usuario que reenvía los contenidos almacenados en su buffer hacia el siguiente eslabón) entre los propios usuarios del sistema que permite reducir la carga del servidor de Streaming del sistema.

Recientes propuestas intentan utilizar el concepto *peer-to-peer* (aparecido originalmente en la distribución de ficheros de música en Internet) para crear un sistema de almacenamiento distribuido de contenidos multimedia que se puede utilizar para implementar un sistema de Streaming totalmente distribuido. En estas arquitecturas, al igual que ocurre con las técnicas de *Chaining*, los Players de los usuarios realizan funciones de servidor de contenidos para las peticiones de otros usuarios del sistema.

6. ARQUITECTURAS DE STREAMING A GRAN ESCALA:

A pesar del atractivo de los servicios de Streaming para el público en general, su implementación hasta el momento no ha sido tan grande como se pudiese esperar, debido a la dificultad de diseñar y construir sistemas de Streaming de gran escala capaz de atender a decenas de miles de peticiones simultáneas.

La construcción de sistemas de video bajo demanda de gran escala está actualmente limitada por dos factores: la capacidad de transmisión simultánea de videos (capacidad de streaming) que puede soportar el servidor y la red de comunicación, y por otro lado, los elevados costos requeridos para su construcción.

Para proveer servicios de Streaming capaz de atender a decenas de miles de usuarios es imprescindible el diseño de sistemas de Streaming a gran escala eficientes y con un costo asumible.

6.1 Requisitos de las arquitecturas de Streaming a gran escala:

A la hora de realizar el diseño de un sistema Streaming a gran escala, además de tener que proporcionar una alta capacidad de streaming, también son de vital importancia considerar los conceptos de escalabilidad, tolerancia a fallos, costo, balanceo de la carga y compartición de recursos, que describimos a continuación:

6.1.1 Escalabilidad del Sistema de Streaming:

En términos generales, la escalabilidad hace referencia a la capacidad del sistema para mantener, sino mejorar, su rendimiento a medida que aumenta el número de clientes. Ninguna instalación de Streaming puede crecer desde 0 a un millón de usuarios de un día para otro. Por lo tanto, sobredimensionar el sistema de Streaming teniendo en cuenta los posibles usuarios futuros, puede dar a lugar a que cuando una capacidad

adicional se requiera el sistema ya resulte obsoleto debido a los avances en la tecnología.

Los sistemas de Streaming deben permitir ajustar su capacidad inicial a los requerimientos de los usuarios para así reducir la inversión inicial. Pero al mismo tiempo deben conservar intacta la capacidad de crecimiento futuro.

La escalabilidad es una de las características más importantes para un sistema Streaming, permitiendo que se pueda ajustar el tamaño inicial del sistema a los requerimientos de los usuarios, pero manteniendo la posibilidad de un fácil crecimiento para poder soportar más usuarios ó nuevos servicios.

En general, en los sistemas distribuidos existen dos tipos de escalabilidad: la vertical y la horizontal. Al hablar de escalabilidad vertical nos referimos a incrementar el número de recursos en la misma máquina para conseguir atender a un mayor número de usuarios. La escalabilidad horizontal consiste en incrementar el número de máquinas que integran el sistema.

6.1.2 Tolerancia a fallos:

Los sistemas de Streaming tienen que continuar dando servicio a los usuarios, incluso si uno o más componentes de la arquitectura fallan. En sistemas de gran escala, normalmente enfocados a un público que paga por unos servicios, no es asumible una interrupción del servicio debido a un fallo de alguno de los componentes. Estas interrupciones de servicio, en caso de producirse, pueden producir un grave perjuicio económico y de imagen. Los sistemas de gran escala son más susceptibles de sufrir algún fallo debido al gran número de componentes (servidores, redes, discos duros) que los forman. Por lo tanto, es importante tener en cuenta la tolerancia a fallos en el diseño de estos sistemas. Una de las técnicas utilizadas para aumentar la tolerancia a fallos, es la replicación de los componentes más críticos del sistema. Aunque reduce la posibilidad de interrupción del servicio, esta técnica

implica una gran inversión en componentes que no estarán en funcionamiento durante la mayor parte del tiempo de vida del componente, y que pueden llegar a quedarse obsoletos incluso antes de ser utilizados. La utilización de componentes de respaldo es muy costoso, por lo cual se suele utilizar cuando no hay más remedio, por ejemplo en sistemas centralizados.

Una alternativa mejor y menos costosa consiste en que los componentes de respaldo formen parte de la propia arquitectura. De esta forma, el posible respaldo para un servidor del sistema es otro servidor de la misma arquitectura. La mejor forma de implementar esta política es logrando que la atención de los usuarios no dependan de un único servidor, sino que puedan ser atendidos por distintos componentes del sistema, en función de las necesidades. La utilización de una arquitectura totalmente distribuida, en la cual la gestión de las peticiones se realiza de forma descentralizada, es la forma más fácil de obtener la tolerancia a fallos sin necesitar componentes de respaldo.

6.1.3 Costo del Sistema de Streaming:

El Streaming a gran escala en el ámbito empresarial está mayormente enfocado a ofrecer servicios de entretenimiento y multimedia a grandes ciudades (si no a países enteros). Este tipo de instalaciones requieren de una gran inversión y, por lo tanto, es imprescindible un control muy estricto de los costos de la arquitectura de Streaming. Desde el punto de vista de la arquitectura de Streaming, los componentes que requieren una mayor inversión son las redes de comunicación y los servidores de video.

El costo de las redes de transmisión depende básicamente del ancho de banda requerido y de la tecnología utilizada. El costo asociado con una red individual se puede incrementar exponencialmente si el ancho de banda requerido es muy grande.

Para reducir el costo del sistema de comunicaciones es recomendable que la arquitectura de Streaming a gran escala no requiera la utilización de redes con anchos de banda muy grandes.

Con respecto a los servidores, los componentes que más influyen en su costo son los asociados con el ancho de banda servicio y el sistema de almacenamiento requerido para soportar el catálogo de contenidos del sistema. El costo del servidor, al igual que ocurre con la redes, depende de la capacidad de servicio deseada. Un servidor más potente necesitará la utilización de técnicas de “clustering” y discos RAID para poder soportar el ancho de banda requerido, aumentando el costo considerablemente.

En general, parece lógico pensar que se obtiene la mejor relación costo/rendimiento utilizando componentes pequeños y cuyo uso esté generalizado en el mercado. Se deben evitar requisitos que necesiten la utilización de componentes demasiado complejos ó que requieran de las últimas tecnologías disponibles en el mercado.

6.1.4 Balanceo de la Carga:

La distribución de la carga entre los distintos servidores del sistema, es importante debido a que las peticiones de los usuarios siguen una distribución no uniforme. Esta característica puede provocar un desbalanceo en el volumen de trabajo de los servidores y una pobre utilización de los recursos globales del sistema. El sistema de Streaming debería permitir que parte de la carga de los componentes (servidores ó redes) más saturados, se pueda desviar a otros componentes menos cargados. Una correcta redistribución de la carga en el sistema, permite reducir la probabilidad de rechazo de servicio a los usuarios.

6.1.5 Compartición de Recursos:

Hoy en día, la posibilidad de poder compartir de recursos entre los usuarios (mediante técnicas de broadcast, multicast, etc.) es la clave para el diseño e implementación de sistemas de Streaming eficientes y con una buena relación costo/prestaciones.

La eficiencia de las técnicas de multicast depende, entre otros factores, del volumen de peticiones que recibe un servidor. Un mayor volumen de peticiones, implica una

mayor probabilidad de que las nuevas peticiones puedan compartir recursos con las peticiones activas. De esta forma, el número de usuarios gestionados por un servidor puede aumentar la eficiencia de las políticas de multicast y por tanto la eficiencia del sistema.

6.2 Alternativas actuales para los sistemas de Streaming a Gran Escala:

La construcción de un sistema de Streaming a gran escala está actualmente limitada tanto por la capacidad del servidor (ancho de banda de servicio) como por la capacidad de transmisiones simultáneas que puede soportar una red de comunicaciones (ancho de banda de la red).

Actualmente, la implementación de un sistema de Streaming a gran escala que pueda soportar un gran número de streams concurrentes, requiere la disposición de varios servidores, que ofrecen la transmisión de video y los servicios de reproducción bajo la forma de un sistema distribuido.

Sin embargo, las distintas aproximaciones para crear esta infraestructura pueden variar desde sistemas completamente centralizados (mediante un cluster de servidores) que utilizan hardware dedicado en la capa de red sin almacenamiento intermedio, hasta sistemas completamente descentralizados que replican todo el catálogo de contenidos en servidores cercanos al usuario final.

Al diseñar un sistema Streaming a gran escala nos podemos basar en las arquitecturas disponibles hoy en día, ya vistas en el apartado anterior. Cada una de estas arquitecturas tiene sus ventajas e inconvenientes dentro de un entorno de servicio a gran escala que comentamos a continuación:

- i. **Sistemas centralizados**: Un sistema de Streaming a gran escala centralizado requiere un servidor y una red principal capaces de soportar un gran ancho de banda. Un servidor con estas características puede llegar a ser muy costoso y complejo de diseñar / construir. La disponibilidad de redes capaces de soportar estos volúmenes de tráfico puede estar limitada por la tecnología disponible.

Además el sistema resultante no es tolerante a fallos, ni escalable ya que el crecimiento del sistema está limitado por la red y el servidor principal.

La única ventaja de esta arquitectura es la alta eficiencia que pueden obtener de las políticas multicast. Al estar todos los usuarios conectados a la misma red, la probabilidad de compartición de streams es la más alta de todas las arquitecturas de Streaming.

Estos sistemas adolecen de una serie de problemas que dificulta su candidatura a una instalación de gran escala.

ii. Servidores independientes: Esta arquitectura no requiere ni redes grandes, ni servidores complejos para lograr la alta capacidad de streaming requerida por los sistemas de gran escala. Estos sistemas permiten una escalabilidad ilimitada ya que para aumentar la capacidad del sistema únicamente se necesita añadir nuevas redes locales (más los servidores correspondientes) al sistema. La principal desventaja de esta arquitectura en un entorno de gran escala son los requerimientos de almacenamiento en los servidores (cada servidor necesita una copia entera del catálogo de contenidos), la compartición de recursos está restringida exclusivamente a los usuarios locales y el balanceo de carga es casi imposible a no ser que las redes independientes estén interconectadas.

iii. Sistemas basados en servidores-proxy de un nivel: Como ya hemos visto, uno de los problemas de estos sistemas estriba en la escalabilidad limitada derivada de la dependencia de unos componentes centralizados como son el servidor y la red principal. La capacidad de crecimiento del sistema dependerá de la capacidad de estos componentes centralizados. De todas formas, disponen de un mayor margen de crecimiento comparado con los sistemas centralizados.

Los requisitos globales de ancho de banda de red en estos sistemas, son mayores debido a que las peticiones que no pueden ser atendidas desde la cache de los servidores-proxy, requieren el doble de recursos de red (red principal + red local)

para poder ser atendidas desde el servidor principal.

Por otro lado, al gestionarse la mayoría de las peticiones en los servidores-proxy, también se reduce la probabilidad de compartición de recursos entre usuarios con respecto a los sistemas centralizados.

iv. Servidores basados en servidores-proxy paralelos/jerárquicos: Básicamente estas arquitecturas tienen los mismos parámetros de rendimiento que los sistemas basados en servidores-proxy de un nivel. La única diferencia estriba en la utilización de un servidor paralelo que aporta múltiples puntos de entrada al servidor y permite reducir el cuello de botella que representa la red principal en los sistemas de servidores-proxy de un nivel.

En última instancia, la escalabilidad de estas arquitecturas depende de la escalabilidad asociada con el servidor paralelo/jerárquico. Por otro lado, parte del ahorro de los costos obtenidos al reducir los requerimientos de almacenamiento se compensan con el costo asociado con el servidor paralelo.

Los principales inconvenientes que tienen estas arquitecturas son la insuficiente escalabilidad y el elevado costo requerido para el sistema. Por lo tanto, existe la necesidad de una nueva arquitectura, específicamente diseñada para un sistema de Streaming a gran escala.

Principales características de las arquitecturas.

Arquitectura	Alta capacidad de servicio	Grandes anchos de banda	Tolerante a fallos	Escalable	Compartición de recursos	Coste
<i>Centralizada</i>	Si	Si	No	No	Si	No
<i>Servidores independientes</i>	Si	No	Si	Si	No	No
<i>Basados en servidores-proxy</i>	Si	Si	~	No	~	~
<i>Distribuidos a nivel del cliente</i>	Si	No	Si	Si	No	No

Figura 2.3: Tabla de las principales características de las arquitecturas.

v. **Sistemas distribuidos a nivel de usuario:** La principal desventaja con la que se tropiezan estos sistemas es el incremento de la complejidad y costo de los componentes de los usuarios, al tener que incorporar buffers más grandes y requerir anchos de bandas de red capaces de soportar más de un stream simultáneo. Además, la eficiencia de las técnicas basadas en “Chiang” ó “peer-to-peer” depende de la existencia de una alta conectividad entre los usuarios del sistema, lo cual puede exigir la utilización de topologías muy costosas (malla, crossbar, etc.) en la conexión de los usuarios al sistema.

En la figura anterior mostramos las principales características de las distintas arquitecturas de Streaming a gran escala. A pesar de que todas estas arquitecturas son capaces de conseguir una capacidad alta de streaming, ninguna de ellas alcanza a cumplir con todas las características deseables para un sistema de Streaming a gran escala.

7. ESTRUCTURA DE UN CANAL DE TELEVISION

El montar un canal de televisión, para llevar la señal que ésta emite hacia los usuarios, conlleva una amplia demanda tanto de hardware como de presupuesto, además de un amplio conocimiento del funcionamiento de las redes y de los servicios necesarios para su correcto funcionamiento. A modo de resumen, cabe destacar, que llevar la señal al espectador (usuario) involucra una arquitectura tradicional cliente/servidor: la computadora del cliente pide una señal y el servidor envía al cliente un flujo de datos conteniendo dicha señal. El proveedor de la señal de televisión necesita enviar a cada usuario un flujo individual, ya que la transmisión se realiza a través de la transmisión Unicast (la cual fue descrita en capítulos anteriores). Cuando se está transmitiendo en vivo a través de Internet, los proveedores están literalmente enviando miles de veces flujos idénticos del servidor a los computadores de los espectadores.

Uno de los objetivos de la investigación es el de lograr transmitir programas radiales y televisivos a los usuarios, minimizando el costo de operación, pero sin descuidar la

eficiencia del sistema. Es por ello que la solución que se planteará a continuación, conlleva un presupuesto no tan elevado, pero que satisface dichos objetivos planteados. Como se desea transmitir a usuarios que tengan reproductores que hoy en día se encuentran en la mayoría de las máquinas, se ha optado por transmitir con los servidores de Streaming “Windows Media Server” y “Real Server”. Al realizar un estudio profundo de las dos tecnologías, se ha descubierto que Real Server puede comportarse como el servidor de Microsoft, Windows Media Server, emulándolo a la perfección, sin la necesidad de tener que instalarlo. Es por ello que solo se ocupará el servidor de RealNetworks, pero igualmente se darán servicios para los dos grandes grupos de usuarios, como son aquellos con los reproductores Windows Media Player y Real Player.

7.1 Proceso de Transmisión de un Canal de Televisión

Los sistemas con los cuales los canales de televisión actuales cuentan, conllevan un gran consumo de recursos, tanto humanos, como de hardware y software. Es por ello, que se propone 2 estructuras o modelos de un sistema de Streaming para un canal de televisión, los cuales se diferencian por un ahorro considerable de hardware y por consecuencia una disminución considerable en los costos. A continuación se explican los dos modelos:

7.1.1 Modelo de Transmisión General

Al igual que la estación de radio, el sistema que permitirá al canal de televisión transmitir a nuestros usuarios, está compuesto por tres módulos claramente definidos: Producción que es el encargado de generar y emitir la señal tanto de audio como video, Codificación que es el encargado de digitalizar la información analógica generada por la producción y transmisión el cual es el encargado de entregar el flujo de datos generados por el canal de televisión a los clientes. Estos módulos están representados en la figura posterior **Figura 7.1**.

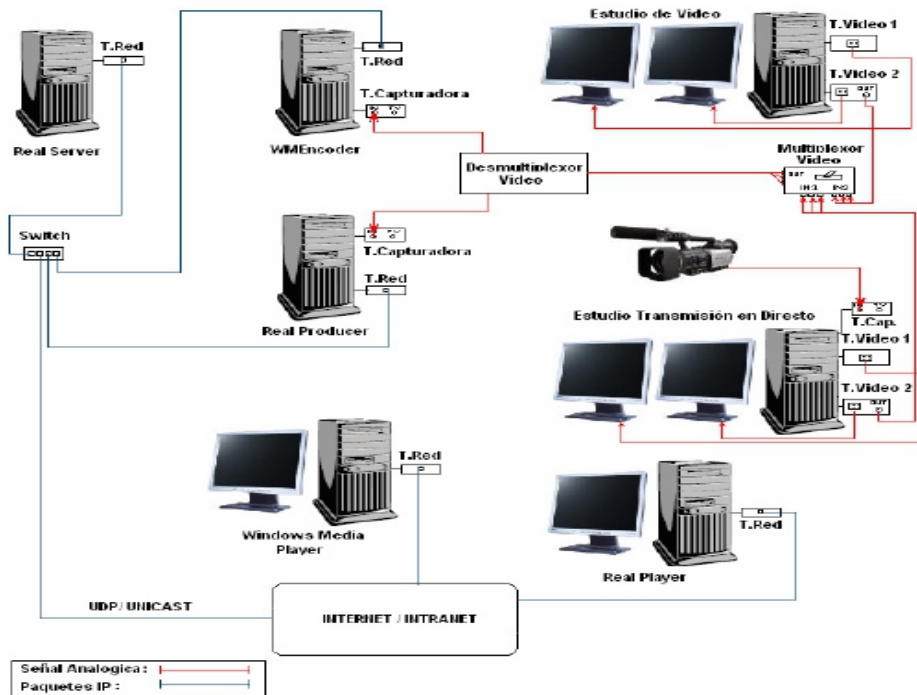


Figura 7.1: Estructura del Canal de Televisión (Modelo General).

En la mayoría de los canales de televisión que operan hoy en día, cuentan con departamentos claramente diferenciados, especialistas en cada uno de los aspectos principales de producción, como lo son el departamento de edición, departamento de Periodismo, departamento de Transmisión Satelital, departamento de Broadcasting, etc., para lo cual cuentan con una amplia gama de recursos en cuanto a infraestructura, insumos, recursos humanos, capital y sistemas de automatización tipo WAN. Es por ello que se debe estudiar muy enfáticamente los objetivos que se desean lograr al desear montar un canal de televisión, y analizar, de acuerdo a éstos objetivos, si es factible y rentable realizar el proyecto. Es por ello que se ha diseñado el sistema actual con un balance entre los factores eficiencia/costos sin descuidar el correcto funcionamiento de nuestro canal de televisión (igualmente para la estación de radio). Como se mencionó anteriormente el proceso de emisión de un canal de televisión está dividido en módulos, los cuales se explican con más detalle:

- i. **Producción:** Al igual que una estación de radio, el canal de televisión también necesita de una producción eficiente, que entregue videos de buena calidad, y de la forma más transparente posible para el usuario. En la

figura se muestra que el proceso de producción, que como se puede observar, consta de dos estudios claramente diferenciados, los cuales son detallados a continuación:

- ii. **Estudio de Video:** La tarea del Estudio de Video es simple, debe de reproducir archivos de video, que ya han sido grabados y editados con anterioridad, y que se quieren entregar en diferido a los usuarios. Para ello se puede utilizar cualquier Reproductor de Video que posea la característica de leer los videos con formato “Windows Media”, “Real Media” y cualquier otro formato. Una recomendación es el de utilizar un reproductor, que además de poseer éstas características, reproduzca además clips en formato DVD, VCD, DviX, etc., ya que en algún momento se podría emitir una película o algún documental proveniente de un dispositivo especial. En lo particular, se ocupara el reproductor “WinDVD” que permite realizar lo antes descrito.

Para ello el estudio de video debe contar con una tarjeta de sonido y dos tarjetas de video, de las cuales una debe tener una salida de televisión la cual deberá ir conectada, junto con la salida de la tarjeta de sonido, al multiplexor de video. La idea de esto, es que el reproductor es manejado desde uno de los escritorios (monitores) y la previsualización del video debe ir en el otro escritorio a través de la tarjeta de video que contiene la salida de televisión, es así como solo la señal de video sale del estudio de video hacia el multiplexor.

- iii. **Estudio de Transmisión en Vivo:** Su función principal es de capturar la señal proveniente de las cámaras filmadoras de video y transmitir las a los codificadores.

Para ello se puede utilizar cualquier software capturador de video, aunque se recomienda utilizar el software por defecto que trae el CD de instalación de la capturadora de video, ya que éste es creado para dicha tarjeta

capturada en especial, y con el otro software correríamos el riesgo de que el sistema se caiga por alguna razón inesperada.

La salida proveniente de los dos estudios anteriormente mencionados, va dirigida un “Multiplexor de Video”, el cual se encarga de seleccionar cuál de las entradas de video es la entregada a los codificadores de video.

El multiplexor su función principal es el de elegir cuál de las dos entradas de video van dirigidas a los codificadores de video. Como se puede observar en la figura, existen dos entradas de video provenientes de dos fuentes distintas, las cuales son canalizadas a la salida de video según sea el interés del administrador. Por lo tanto el multiplexor de video cuenta con un Switch que selecciona una de ellas como salida.

- iv. **Codificación:** Cuando la señal sale del multiplexor de video, independientemente de su estudio de procedencia, debe ingresar a los codificadores, para que éstos entreguen la señal digitalizada al servidor de Streaming. Su funcionamiento es similar al de la estación de radio, con la diferencia que la señal es una transmisión de video, la cual ingresa al codificador por una tarjeta capturadora. En cuanto a la comunicación de éste con el servidor, se debe especificar el tipo de compresión tanto de video como de sonido, dependiendo del CODEC seleccionado

- v. **Transmisión:** Éste proceso es exactamente igual al de una estación de radio, el servidor de Streaming, recibe el flujo de datos digitalizado por el codificador, y lo retransmite a los usuarios con la calidad configurada en el ítem anterior.

7.1.2 Modelo de Transmisión Reducido

Como se puede apreciar, en el modelo anterior se necesita contar con hardware específico que realice la tarea de generar la señal transmitida, a partir de un estudio

especializado en la edición y reproducción de material almacenado, y a partir también de un estudio encargado de recoger la señal en vivo transmitida de cualquier lugar geográfico. Por lo tanto, para lograr que el canal de televisión sea factible de realizar, se debe contar con un modelo reducido, que cuente con lo esencial, pero sin perjudicar la calidad de transmisión. Al igual que en el modelo anterior, el proceso de transmisión está dividido en módulos claramente diferenciados, los cuales están representados en la siguiente figura:

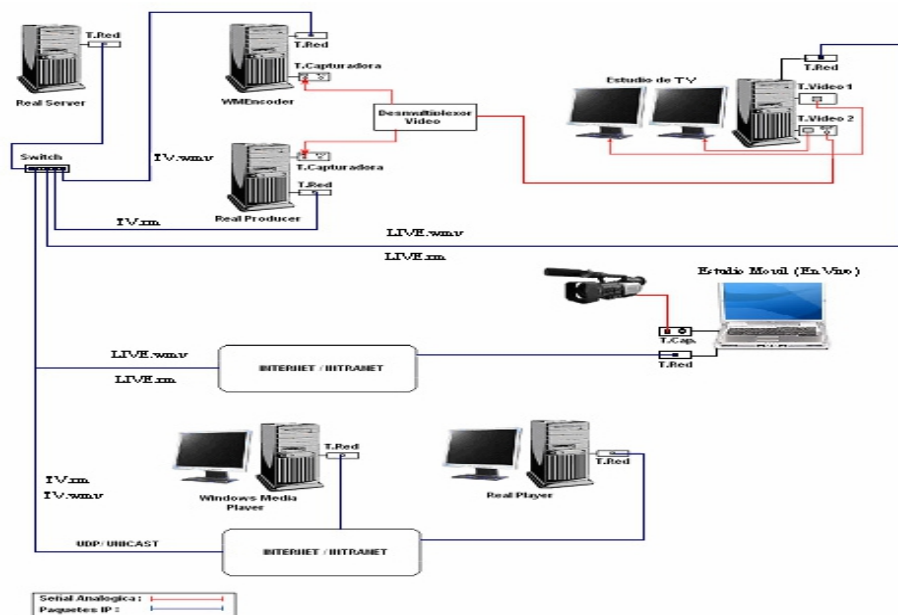


Figura 7.2: Estructura del Canal de Televisión (Modelo Reducido).

Cabe destacar, que el modelo que se plantea es una solución lo más reducida, para ahorrar recursos y así abaratar el costo total del sistema, sin descuidar el objetivo principal, el cual es entregar un medio audiovisual a los usuarios de buena calidad y de forma eficiente, el proceso de emisión de un canal de televisión está dividido en módulos, los cuales se explican con más detalle:

- i. **Producción:** A diferencia del modelo anterior, el módulo de producción cuenta con la “Estación de TV” y la “Estación Móvil”. La “Estación de TV” es la encargada de automatizar el proceso de transmisión, seleccionando un video

almacenado en disco o simplemente mostrando una transmisión en vivo realizada desde cualquier lugar del mundo (Estación Móvil), a través de un codificador de Streaming (“Real Producer” o “Windows Media Encoder”). Para ello la “Estación Móvil” debe contar con cualquier computador con tarjeta capturadora y que tenga acceso al servidor (Real Server), para que le envíe la señal en vivo. Dicha señal en vivo será tomada por la “Estación de TV” desde el servidor y será retransmitida a los usuarios.

Una vez que la señal es generada por la “Estación de TV”, la salida de su tarjeta de video (salida analógica) es dirigida a los codificadores los cuales son los encargados de digitalizar la señal.

- ii. Codificación:** Cuando la señal sale de la “Estación de TV”, debe ingresar a los codificadores, para que éstos entreguen la señal digitalizada al servidor de Streaming. Su funcionamiento es similar al del método anterior, por lo que la señal entregada por la “Estación de TV” pasa directamente a los codificadores “Windows Media Encoder” y al “Real Producer”. En cuanto a la comunicación de éste con el servidor, se debe especificar el tipo de compresión tanto de video como de sonido, dependiendo del CODEC seleccionado.

- iii. Transmisión:** Éste proceso es exactamente igual al de una estación de radio y al método anterior, el servidor de Streaming, recibe el flujo de datos digitalizado por el codificador, y lo retransmite a los usuarios, con la calidad configurada en el ítem anterior.

8. LEER UN STREAM DE UN VIDEO FORMATO AVI DIVIDIRLO EN FRAMES

8.1 Breve reseña histórica del formato AVI:

El formato *AVI* fue definido por Microsoft para su tecnología “Video for Windows” en 1992. Posteriormente fue mejorado mediante las extensiones de formato del grupo OpenDML de la compañía Matrox. Estas extensiones están soportadas por Microsoft, aunque no de manera oficial, y son denominadas **AVI 2.0**.

8.2 Cómo funciona el formato AVI:

El formato *avi* permite almacenar *simultáneamente* un flujo de datos de video y varios flujos de audio. El formato concreto de estos flujos no es objeto del formato AVI y es interpretado por un programa externo denominado códec. Es decir, el audio y el video contenidos en el AVI pueden estar en cualquier formato. Por eso se le considera un formato contenedor.

Para que todos los flujos puedan ser reproducidos simultáneamente es necesario que se almacenen de manera intercalada (*interleave*). De esta manera, cada fragmento de archivo tiene suficiente información como para reproducir unos pocos fotogramas junto con el sonido correspondiente.

Obsérvese que el formato AVI admite varios flujos de datos de audio, lo que en la práctica significa que puede contener varias bandas sonoras en varios idiomas. Es el reproductor multimedia quien decide cuál de estos flujos debe ser reproducido, según las preferencias del usuario.

Los archivos AVI se dividen en fragmentos bien diferenciados denominados *chunks*. Cada *chunk* tiene asociado un identificador denominado *etiqueta FourCC*. El primer fragmento se denomina *cabecera* y su papel es describir *meta-información* respecto al archivo, por ejemplo, las dimensiones de la imagen y la velocidad en fotogramas por

segundo. El segundo *chunk* contiene los flujos entrelazados de audio y video. Opcionalmente, puede existir un tercer *chunk* que actúa a modo de índice para el resto de *chunks*.

8.3 Cómo se reproduce un archivo AVI

Para reproducir un archivo AVI es necesario lo siguiente:

- Un reproductor de video capaz de interpretar el formato AVI.
- El *códec* de video para interpretar el flujo de video.
- El *códec* de audio para interpretar el flujo de audio.

La etiqueta FourCC permite identificar el *códec* necesario para interpretar un flujo de audio o video. Cada codec tiene asociados el conjunto de etiquetas que es capaz de reproducir. De esta manera, el reproductor de video es capaz de elegir el *codec* pertinente sin intervención del usuario.

El reproductor consecutivamente lee fragmentos del archivo AVI. Después separa cada uno de los flujos de audio y video que se encuentran intercalados (interleave) en el archivo. Cada uno de estos flujos, una vez separados, se almacena en un buffer de memoria y se pasan al *codec* correspondiente. El *códec* de video devuelve otro *buffer* que contiene cada uno de los fotogramas a reproducir. El *códec* de audio retorna otro *buffer* con la muestra digital de sonido a reproducir. Con esta información, el reproductor solamente tiene que sincronizar los fotogramas y el sonido y reproducirlos a la velocidad adecuada.

En el sistema operativo Mac OS es perfectamente posible visualizar archivos AVI, siempre que los codecs utilizados estén soportados por quicktime, bien directamente o a través de plugins. Existen otros reproductores de archivos multimedia para esta plataforma que también permiten visualizar correctamente archivos AVI.

8.4 Código en Visual Studio 6.0 de extracción de un video Avi y dividirlo en Frames:

```
//// test1.cpp : Defines the entry point for the console application.

#include "stdafx.h"

#include <windows.h>

#include <assert.h>

#include <string.h>

#include "vfw.h"

#pragma comment( lib, "vfw32.lib" )

BOOL CreateFromPackedDIBPointer(LPBYTE pDIB, int iFrame) //esta función creará el bitmap a
partir del Frame

{

if(pDIB==NULL) return 0

//Creates a full-color (no palette) DIB from a pointer to a full-color memory DIB

//get the BitmapInfoHeader

BITMAPINFOHEADER bih;

RtlMoveMemory(&bih.biSize, pDIB, sizeof(BITMAPINFOHEADER)); //now get the bitmap
bits

if (bih.biSizeImage < 1)

{

return FALSE;

}

BYTE *Bits = new BYTE[bih.biSizeImage];

RtlMoveMemory(Bits, pDIB + sizeof(BITMAPINFOHEADER),bih.biSizeImage);

//and BitmapInfo variable-length UDT

BYTE memBitmapInfo[40];

RtlMoveMemory(memBitmapInfo, &bih, sizeof(bih));
```

```
BITMAPFILEHEADER bfh;

bfh.bfType=19778; //BM header

    bfh.bfSize=55 + bih.biSizeImage;

    bfh.bfReserved1=0;

    bfh.bfReserved2=0;

bfh.bfOffBits=sizeof(BITMAPINFOHEADER) +sizeof(BITMAPFILEHEADER); //54

    CString FileName;

    FileName.Format("c:\\frame\\Frame-%05d.bmp", iFrame);

    FILE* fp=fopen(FileName, "wb");

    if (fp!=NULL)

        {

            fwrite(&bfh, sizeof(bfh), 1, fp);

            fwrite(&memBitmapInfo, sizeof(memBitmapInfo), 1, fp);

            fwrite(Bits, bih.biSizeImage, 1, fp);

            fclose(fp);

        }

    else {

        TRACE0(_T("Error writing the bitmap file"));

        return FALSE;

    }

    delete [] Bits;

return TRUE;

}

BOOL ExtractAVIFrames(CString szFileName) // a esta función se le pasa en el atributo
szFileName la ruta de la ubicación del video .avi en el sistema de archivos, por ejemplo
'C:\\temp\\video.avi'

{

    AVIFileInit(); //llamada a función de inicialización

    PAVIFILE avi;
```

```
//se crea la variable avi

int res=AVIFileOpen(&avi, szFileName, OF_READ, NULL); //llamada a la apertura del fichero

if (res!=AVIERR_OK) //aquí se comprueba si el proceso de apertura ha finalizado exitosamente o
no

{
//an error occurs
if (avi!=NULL)
    AVIFileRelease(avi);
    return FALSE;
}

AVIFILEINFO avi_info; //avi_info va a representar la información recogida en el .avi
AVIFileInfo(avi, &avi_info, sizeof(AVIFILEINFO)); //avi_info se carga con la información
contenido en el fichero .avi. No olvidar que se está pasando avi_info por referencia, lo indica el
símbolo &
CString szFileInfo; //creación del objeto szFileInfo de tipo CString
szFileInfo.Format("Dimention: %dx%d\n"
"Length: %d frames\n"
"Max bytes per second: %d\n"
"Samples per second: %d\n"
"Streams: %d\n"
"File Type: %d", avi_info.dwWidth,
avi_info.dwHeight,
avi_info.dwLength,
avi_info.dwMaxBytesPerSec,
(DWORD) (avi_info.dwRate / avi_info.dwScale),
avi_info.dwStreams,

avi_info.szFileType); //Se le asignan una serie de cadenas de texto a szFileInfo en base a lo
contenido en avi_info, es información de la codificación del .avi

AfxMessageBox(szFileInfo, MB_ICONINFORMATION | MB_OK);
PAVISTREAM pStream; //se define el objeto que va a representar el Stream
res=AVIFileGetStream(avi, &pStream, streamtypeVIDEO /*video stream*/, //Se llama a la
función de vfw para volcar el contenido del .avi en el Stream
0 /*first stream*/);
if (res!=AVIERR_OK) //Se comprueba que el proceso de volcado haya finalizado correctamente.

{
if (pStream!=NULL) AVIStreamRelease(pStream);
    AVIFileExit();
return FALSE;
}
```

```
}

//do some task with the stream
int iNumFrames;
int iFirstFrame;
iFirstFrame=AVIStreamStart(pStream); //A continuación va a comenzar el tratamiento del primer
Frame del Stream

if (iFirstFrame===-1) //y se controla la no aparición de errores en el tratamiento del Stream
{
//Error getting the frame inside the stream
if (pStream!=NULL)
    AVIStreamRelease(pStream);
    AVIFileExit();
    return FALSE;
}

iNumFrames=AVIStreamLength(pStream); //Se obtiene el número total de Frames para poder
manejar su procesamiento en el bucle

    if (iNumFrames===-1)
    {
//Error getting the number of frames inside the stream
if (pStream!=NULL) AVIStreamRelease(pStream);
AVIFileExit();
return FALSE;
}
//getting bitmap from frame
BITMAPINFOHEADER bih;
ZeroMemory(&bih, sizeof(BITMAPINFOHEADER));
bih.biBitCount=24; //24
bit per pixel
bih.biWidth = avi_info.dwWidth;
bih.biHeight = avi_info.dwHeight;
bih.biClrImportant=0;
bih.biClrUsed = 0;
bih.biCompression = BI_RGB;
bih.biPlanes = 1;
bih.biSize = 40;
bih.biXPelsPerMeter = 0;
bih.biYPelsPerMeter = 0;
//calculate total size of RGBQUAD scanlines (DWORD aligned)
bih.biSizeImage = (((bih.biWidth * 3) + 3) & 0xFFFC) * bih.biHeight ;
PGETFRAME pFrame;
```

```
pFrame=AVIStreamGetFrameOpen(pStream,

(BITMAPINFOHEADER*) AVIGETFRAMEF_BESTDISPLAYFMT);

//Si todo ha ido bien con el primer Frame se desarrolla el tratamiento del resto de Frames
int index=0;
for (int i=iFirstFrame; i<iNumFrames; i++) //desde el primer Frame hasta el número total de
Fr2ames, incrementando de uno en uno...

{
    index= i-iFirstFrame;
    BYTE* pDIB = (BYTE*) AVIStreamGetFrame(pFrame, index);
    //Se vuelca el Stream en pDIB  CreateFromPackedDIBPointer(pDIB,
    // index);
    //llamada a la función que viene a continuación y que trata el Frame
    }
    //A continuación se procede al cierre del Stream tras el tratamiento
    AVIStreamGetFrameClose(pFrame);

if (pStream!=NULL)

    AVIStreamRelease(pStream);

    AVIFileExit();

    return TRUE;

}

int main(int argc, char* argv[])

{

    ExtractAVIFrames("C:\\prueba.avi");

    return 0;

}
```