

72



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

"EVALUACION DE TECNICAS DE COMPRESION DE
AUDIO Y VIDEO DIGITAL A TRAVES DE INTERNET".

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A :

GUSTAVO MENDOZA MORON

DIR. TESIS: ING. JUAN JOSE CARREON GRANADOS



MEXICO, D. F. CIUDAD UNIVERSITARIA

NOVIEMBRE 2002

TESIS CON
FALSA EL OR.GEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Autorizo a la Dirección General de Bibliotecas de la
UNAM a difundir en formato electrónico e impreso el
contenido de mi trabajo recepcional.

NOMBRE: Gustavo Mendoza

Morón

FECHA: Nov 8 2002

FIRMA: 

AGRADECIMIENTOS

Gracias:

A mi madre por apoyarme durante toda la carrera y haberme dado la orientación necesaria para lograr está meta.

A mis hermanos por apoyarme en los momentos más difíciles.

A toda mi familia por estar siempre al pendiente de mis estudios.

A la Universidad y a la Facultad de Ingeniería por darme los conocimientos necesarios para mi formación profesional.

A mi director de tesis el ing. Juan José Carreón Granados por brindarme su apoyo al desarrollar este tema.

A los profesores que compartieron sus conocimientos conmigo.

A mis amigos de la Facultad por su ayuda incondicional y su amistad.

Índice

Introducción.....	1
Capítulo 1 Antecedentes.....	3
1.1 Introducción a Multimedia.....	4
1.2 Internet y las Comunicaciones Multimedia	4
1.3 Medios Discretos y Continuos.....	5
1.4 Señales Digitales.....	5
1.4.1 <i>Sampling</i> (muestreo).....	6
1.4.2 Cuantización y Codificación.....	6
1.4.3 Bit Rate (tasa de bits).....	7
1.5 Imágenes Fijas.....	8
1.6 Texto y Gráficos.....	9
1.7 Gráficos e Imágenes en Movimiento.....	10
1.8 Codificación y Decodificación.....	11
1.9 Ancho de Banda vs. Compresión.....	13
1.9.1 Eligiendo un Bit Rate apropiado.....	16
1.9.2 Bit Rate en Audio.....	16
Capítulo 2 Redes Multimedia.....	18
2.1 Aplicaciones multimedia distribuidas.....	19
2.2 Comunicaciones Múltipleer y Peer to Peer.....	20
2.3 Parámetros de rendimiento de redes para multimedia	21
2.3.1 Throughput.....	21
2.3.2 Tasa de error (Error Rate).....	22
2.3.3 Delay.....	23

2.3.4 Round-trip Delay.....	24
2.3.5 Variación de Delay o Jitter.....	24
2.4 Características de las Fuentes de Tráfico Multimedia.....	25
2.4.1 Variación Throughput en el Tiempo.....	25
2.4.2 Dependencia del Tiempo.....	26
2.4.3 Simetría Bidireccional.....	26
2.5 Factores que afectan el rendimiento de la red.....	27
2.5.1 Factores de rendimiento Throughput.....	27
2.5.2 Cuestiones en el rendimiento de error en red.....	29
2.6 Requisitos para tráfico multimedia para redes.....	32
2.6.1 Requisitos Throughput.....	32
2.6.2 Requisitos de confiabilidad en control de error.....	33
2.6.3 Requisitos de Retardo.....	33
2.7 Calidad de Servicio.....	34
<hr/>	
Capítulo 3 Métodos de Codificación Básicos.....	37
3.1 Introducción a los métodos de compresión.....	38
3.2 Métodos de Codificación Básicos.....	42
3.2.1 Codificación Run-Lenght.....	42
3.2.2 Codificación Huffman.....	43
3.2.3 Codificación Aritmética.....	45
3.2.4 La Transformada Coseno Discreta.....	46
3.2.5 Differential Pulse Code Modulation.....	49
3.2.6 Predicción Compensada de Movimiento.....	50
<hr/>	
Capítulo 4 Compresión de Video.....	52
4.1 Compresión de video.....	53
4.1.1 Técnicas sin Estándar.....	53
<hr/>	

4.1.2	JPEG.....	54
4.1.3	H.261.....	59
4.1.4	MPEG-1 Video.....	63
4.1.5	MPEG-2 Video.....	67
4.1.6	MPEG-4 Video.....	69
4.1.7	H.263.....	70
4.1.8	Compresión de imágenes Wavelet.....	70
4.1.9	Compresión de Imágenes Fractal.....	73
<hr/>		
Capítulo 5 Compresión de Audio.....		77
5.1	Compresión de Audio.....	78
5.1.1	Variaciones de Pulse Code Modulation.....	78
5.1.2	MPEG-1 Audio.....	79
5.1.3	Codificador de Audio Perceptual.....	83
<hr/>		
Capítulo 6 Flujos de Audio y Video en Internet.....		84
6.1	Flujos de audio y video en la World Wide Web.....	85
6.2	Streaming video en Internet.....	87
6.2.1	Definiciones básicas.....	87
6.2.2	Definiciones de servidor.....	88
6.2.3	Formatos de archivo de video para Internet.....	89
6.2.4	Configurando una computadora para ver video streaming.....	90
6.3	Trabajo en Internet del Audio.....	90
6.3.1	Tecnología Webcasting.....	91
6.3.2	La transmisión de Internet.....	93
6.3.3	IP Multicast.....	94
6.3.4	Protocolo Streaming de Tiempo Real.....	96
6.3.5	El método Microsoft.....	97
<hr/>		
Conclusiones.....		98
<hr/>		
Bibliografía.....		100
<hr/>		
Glosario.....		101
<hr/>		

Introducción

El acceso universal a la información multimedia es actualmente la principal motivación detrás del diseño de la siguiente generación de computadoras y redes de comunicación. Además de esto, están siendo desarrollados productos para ampliar las capacidades en todas las conexiones de red existentes para soportar tráfico multimedia. La rápida evolución de estas redes ha llegado por los nuevos avances tecnológicos, el aumento de las expectativas públicas y las lucrativas oportunidades empresariales.

En los últimos años se ha hecho popular el uso de métodos de compresión para llevar a cabo la distribución multimedia en Internet. Los algoritmos que se usan en la compresión pueden comprimir señales de audio y video sin pérdidas perceptibles para el oído y la vista humana. Esto ha hecho posible el almacenamiento de grandes cantidades de música y video en pequeñas cantidades de medios de almacenamiento. También ha hecho posible la implementación de la reproducción en tiempo real de música y video sobre redes de computadoras en anchos de banda bajos.

La transmisión y recepción de imágenes y sonidos de manera continua a través de una red se conoce como "*streaming*", esta tecnología permite apreciar el contenido conforme se va teniendo acceso a la información del archivo. Antes se tenía que bajar completamente un archivo de audio o video al disco duro, para después ser visto a través de un reproductor de software. Esto a menudo tomaba mucho tiempo, ya que dependía del tamaño del archivo y del ancho de banda utilizado por el usuario.

La tecnología empleada en este tipo de envío de información no es adecuada para aplicaciones en tiempo real. Lo cual hizo necesario el diseño de nuevos protocolos de red exclusivos para aplicaciones multimedia en tiempo real y de nuevos métodos de compresión, que permitirían el libre acceso a la información cuando fuese requerida. Tal innovación tecnológica surge entonces de las limitaciones técnicas que se tenían hasta entonces y que actualmente se siguen

perfeccionando. De aquí que los métodos de compresión jueguen un papel muy importante, ya que han permitido hacer un uso más eficiente del ancho de banda de Internet y además han mejorado la calidad de audio y video que recibe el usuario final, la evaluación de dichos métodos aplicados a la transmisión de audio y video digital a través de Internet es el objetivo de esta tesis.

Antecedentes

3

Antecedentes

1.1 Introducción a Multimedia

Para mucha gente, multimedia significa la representación de información en términos de texto, audio, video, etc, pero es más que eso. Esto no sólo connota la habilidad de representar varios modos de información sino también la capacidad de manipular y controlar está información por medio de una computadora, así como también la transformación de la información a través de canales de telecomunicación. Multimedia trata entonces la representación de modos de información mezclados—texto, datos, imágenes, audio y video—como señales digitales. Las comunicaciones multimedia tratan la tecnología requerida para manipular, transmitir y controlar estas señales audiovisuales a través de un canal de comunicaciones en red.

1.2 Internet y las Comunicaciones Multimedia

Cualquiera que haya navegado en Internet con un navegador Web ya ha experimentado las comunicaciones multimedia. La mayoría de los documentos HTML en la red involucran tanto texto como imágenes. En unas aplicaciones también hacen uso de diálogos, audio y video. Si se ha tenido experiencia en el uso de Internet, se está conciente de las deficiencias de Internet para comunicaciones multimedia. Es fácil darse cuenta que es mucho más tardado bajar una imagen que bajar solamente texto.

Por ejemplo, analizando el despliegue de una página de Internet que contenga texto e imágenes. Las partes de texto se muestran más rápido que las imágenes. ¿Por qué sucede esto? Una respuesta simple es que las fotos o imágenes requieren mucho más espacio de datos que representar comparados con el texto. Las imágenes en movimiento requieren aún más espacio de datos. Antes de la llegada de la World Wide Web, la mayoría de los usos de Internet involucraba aplicaciones solamente de texto o datos, tales como el e-mail o la transferencia de

archivos (FTP). Las comunicaciones en tiempo real no eran una gran preocupación y bajar gráficos e imágenes eran relativamente poco comunes. Más aún Internet en sus inicios fue raramente usado para comunicaciones habladas.

La World Wide Web ha cambiado todo esto. Internet actualmente es como un camino viejo y sucio, es inadecuado para automóviles de alta velocidad. Hoy en día muchas ciudades han planeado construir nuevas carreteras de información para Internet del futuro, en las cuales las comunicaciones multimedia serán rápidas y eficientes. Sin embargo el diseño de estas supercarreteras de la información es complejo. No simplemente involucra construir redes de banda ancha. Cuestiones tales como el acceso, control y monitoreo deben tratarse.

1.3 Medios Discretos y Continuos

Literalmente, multimedia se refiere a dos o más medios, éstos medios se dividen en dos clases: continuos y discretos. Los medios continuos también llamados temporales cambian con el tiempo. Los medios discretos son independientes del tiempo. Ejemplos de medios continuos son el audio y el video. Ejemplos comunes de medios discretos son texto, formateado o no formateado, imágenes sin movimientos y gráficos.

Es relativamente sencillo el tratar con medios de comunicación discretos. Un archivo de texto que no varía en el tiempo no cambiará si es mandado a Internet ahora o diez minutos después. Sin embargo, este no es el caso con medios continuos. Una transmisión de televisión en vivo en tiempo real contiene información de medios de audio y video, ambos son dependientes del tiempo. El flujo de datos de video y audio deben transmitirse de tal forma que lleguen juntos en sincronía. Si el canal de comunicación permite al flujo de audio que llegue más rápido que el flujo de video, a través del canal, el resultado final de la recepción será perceptiblemente desagradable.

Tanto los medios de comunicación discretos como los continuos dan lugar a requerimientos específicos en los sistemas de telecomunicación usados para transportar dicha información.

1.4 Señales Digitales

En sistemas de telecomunicación tradicionales, la información se transmite generalmente en forma análoga, representada como una señal continua en el tiempo. Sin embargo en las comunicaciones por computadora, el método preferido es transmitir señales audiovisuales en forma digital. Estas señales digitales son

convertidas de su representación análoga original mediante dos procesos: *sampling* y *cuantización* + *encoding*.

1.4.1 *Sampling* (muestreo)

Muestrear una señal en el tiempo significa que la señal análoga es examinada y medida a intervalos T de tiempos regulares. De esta manera una versión muestreada de un señal continua $s(t)$ es representada por estos valores:

$$s(t) = \{s(T), s(2T), s(3T), \dots, s(nT)\}$$

donde T es el intervalo de muestreo y la frecuencia de muestreo es $f = \frac{1}{T}$.

De acuerdo al teorema del muestreo de Nyquist, si una señal contiene componentes de frecuencia arriba de algunas frecuencias f_0 , entonces la frecuencia de muestreo mínima que debe ser empleada para representar la señal exacta es $2f_0$. Por ejemplo si una señal de voz tiene una frecuencia máxima de 3Khz, entonces la frecuencia mínima de muestreo debe ser al menos de 6Khz. Esto es la señal de voz debe ser muestreada por lo menos 6000 veces por segundo para una representación fiel de la señal análoga de voz original.

1.4.2 Cuantización y Codificación

Después de que la señal análoga es muestreada, el valor muestreado es después cuantizado y codificado como una cadena de bits. La cuantización significa la representación de los valores muestreados en términos de un grupo discreto de valores de amplitud. El número de bits usados para representar este valor muestreado determina la exactitud del proceso de cuantización/codificación. Por ejemplo si un valor muestreado $s(iT)$ es cuantizado por un código de 3 bits, si $s(iT)$ es capaz de asumir sólo ocho valores discretos diferentes, desde { 000 ... 111}.

Cada uno de estos valores discretos puede después ser codificado para facilitar la economía en transmisiones o almacenaje de información digitalizada. En la figura 1-1 podemos ver que una señal análoga $s(t)$ es muestreada en puntos discretos en el tiempo nT y después cuantizada y codificada como lo representan las líneas verticales oscuras en cada intervalo de muestreo.

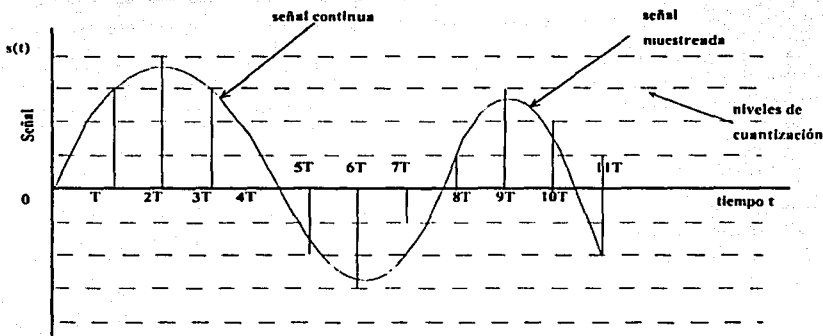


Figura 1-1 Señal analógica muestreada y cuantizada

En aplicaciones de voz en teléfonos, el estándar actual para digitalizar la voz humana es usar 16 bits por muestra, lo cual nos lleva a 2^{16} o 65536 niveles de amplitud distintos que pueden representarse. En otras aplicaciones de compresión de diálogos, a lo mejor solo se requieren cuantizaciones de 8 bits, lo cual significa que nosotros solo podemos distinguir 256 amplitudes diferentes.

1.4.3 Bit Rate (tasa de bits)

La tasa de bits se define como el producto de la tasa de muestreo y el número de bits usados en el proceso de cuantización, por ejemplo, en la voz telefónica, el ancho de banda de la señal de voz es de alrededor de 3 kHz, lo que significa que la tasa de muestreo (frecuencia), de acuerdo al teorema de Nyquist debe ser de al menos 6 kHz. Asumamos una tasa de muestreo más conservativa de 8 kHz. Si nosotros usamos un cuantizador de 8 bits entonces

la tasa de bits requerida para una conversación telefónica = $8000 \times 8 = 64$ kbps

Como otro ejemplo considere el caso de un disco compacto de audio (CD) en audio de alta fidelidad, el ancho de banda es generalmente aceptado como de 20 kHz, así que la mínima frecuencia de muestreo es de 40 kHz. Usando los estándares de la industria de una tasa de muestreo de 44.1 kHz y una cuantización de 16 bits, y para estéreo (2 canales) llegamos a:

la tasa de bits para un disco compacto de audio = $44100 \times 2 \times 16 = 1410$ kbps

En las comunicaciones de datos, la tasa de bits es un parámetro importante. La capacidad del canal de redes de datos públicas es la mayoría de las veces dadas en kilobits o megabits por segundo. Por ejemplo, en la red digital de servicios integrados (ISDN) la tasa de bits estándar empleada para voz es de 64 kbps.

En la tabla 1-1, mostramos las relativas tasas de bits para conversación telefónica, conversación en tele conferencia (que requiere mayor calidad de voz), discos compactos de audio y cinta de audio digital (DAT).

Formato	Tasa de Muestreo (kHz)	Ancho de Banda (kHz)	Rango de Frecuencia (Hz)	Bit rate (kbps)
Telefonía	8.0	3.0	200 - 3,200	64
Tele conferencia	16.0	7.0	50 - 7,000	256
Disco compacto	44.1	20.0	20 - 20,000	1,410
Cinta de Audio Digital (DAT)	48.0	20.0	20 - 20,000	1,536

Tabla 1-1 Formatos de Audio Digital

1.5 Imágenes Fijas

Antes, aludimos el hecho de que las imágenes fijas toman mucho más tiempo para transmitirse que el texto. La razón básica es que las imágenes están compuestas de *pixels* (píxeles), y hay un gran número de éstos en cada imagen. ¿Qué es un píxel? Como no puedes encontrar una definición de píxel en la mayoría de los diccionarios estándares, un buen diccionario técnico define *píxel* como:

Píxel: (sustantivo) la unidad simple más pequeña o punto de una imagen cuyo color o brillo puede ser controlado.

Si nosotros examinamos, debajo de un microscopio, una fotografía en una página de periódico, vemos que la foto está hecha de una trama muy fina de puntos, cada punto tiene un tono específico de gris, de un rango desde el blanco al negro. Estos puntos son píxeles y nosotros vemos que un píxel puede tomar un valor en particular—en el caso de la foto en el periódico—, un tono gris.

En fotos de colores o en pantallas, el píxel puede asumir también valores asociados con el color y el brillo. Una mejor definición de píxel sería:

Píxel: Elemento de dibujo. La más pequeña unidad de área posible de una imagen en una pantalla o guardada en la memoria. Cada *píxel* en una imagen

monocromática tiene su propia brillantez, desde el cero para el negro hasta el máximo valor (ejemplo, 255 para un píxel de 8 bits) para el blanco. En una imagen de color, cada píxel tiene su propia brillantez y color.

Las imágenes por computadora son mapas de bits hechos de *píxeles*. En una resolución estándar para la pantalla de la computadora hay 768 líneas, cada línea contiene 1924 píxeles. Para una exposición a color, suponga que cada valor de color y brillo en cada píxel es especificado por 24 bits (bits x píxel o bpp), entonces el número total de bits para una imagen en la pantalla de la computadora es:

$$\text{Número de bits} = 1024 \times 768 \times 24 = 18.874 \text{ megabits}$$

Por ejemplo supongamos que necesitamos enviar esta imagen en un MODEM a 28.8 kbps. Entonces requerirá

$$\text{Tiempo de transmisión} = \frac{18,874,000}{56,600} = 333.46 \text{ segundos} = 5.55 \text{ minutos.}$$

Esto claramente es demasiado tiempo para que el usuario espere si se está trabajando en tiempo real. ¿Qué se puede hacer al respecto? Hay cuatro métodos básicos:

- 1) **Enviar una imagen a través de un canal más rápido**, como una línea T1 con una velocidad de 1.544 Mbps.
- 2) **Reducir el número de bits por píxel**, permitiendo de esta manera menores niveles discretos para el brillo y menos tonos en el color.
- 3) **Reducir la resolución de la pantalla** (menos píxeles por línea y menos líneas por imagen).
- 4) **Quitar la redundancia en el despliegue**, que significa remover los excesos de píxeles que representan esencialmente el mismo objeto

Las técnicas de compresión de imágenes combinan los métodos 2, 3 y 4.

1.6 Texto y Gráficos

Comparando a las imágenes, el texto plano o el texto formateado requieren mucho menos capacidad de transmisión. Los caracteres del texto plano son representados por 8 bits, o un byte. Los caracteres de texto formateado son representados por 2 bytes. Para una sola página de texto hay 64 líneas y 80 caracteres por línea. Así, una sola página de texto contiene:

$$\text{Número de bits} = 80 \times 64 \times 2 \times 8 = 82 \text{ kbits}$$

la cuál requiere (usando un Modem a 56.6 kbps) sólo 1.4 segundos para transmitirse. En contraste a las imágenes, los gráficos son dibujos hechos por humanos o computadoras, hechos de líneas en espacios. Nosotros podemos considerar un gráfico como una composición de objetos que representan información. Estos objetos pueden ser descritos matemáticamente o por comandos computacionales.

Así una línea recta puede ser descrita en términos de dos puntos finales, un círculo por medio de la localización de su centro y radio, etc. Los gráficos son corregibles o editables, las imágenes no. Los gráficos requieren mucho menos espacio de almacenaje en memoria, que lo requiere una imagen de mapa de bits. Finalmente, los gráficos necesitan mucho menos tiempo para transmitirse a través de una red que lo que una imagen de mapa de bits requiere. En las comunicaciones multimedia es necesario tratar con imágenes y gráficos.

1.7 Gráficos e Imágenes en Movimiento

Las películas de cine están compuestas de secuencias temporales de imágenes gráficas, cada una de las cuáles es llamada *frame* (*marco*). Los objetos gráficos en cada frame varían insignificadamente al frame anterior, así cuando la secuencia de frames es proyectada en el tiempo, el resultado muestra un objeto que es percibido en movimiento. La velocidad de la proyección en término del número de frames por segundo es llamada *frame rate*. Si el frame rate es muy lento, el resultado en el movimiento del objeto es desigual. Las películas visualmente tiene un frame rate entre 25 y 30 frames por segundo (fps). La experiencia muestra que una frame rate de 16 o más es necesaria para representar movimiento fluido.

Las películas de cine y las videocámaras tienen imágenes de objetos en movimiento del mundo real. Cada cámara fotografía un frame individual. Una secuencia temporal de fotografías tomadas a lo largo del tiempo muestra el movimiento de los objetos. En despliegue de video el frame rate es la tasa a la cual la secuencia temporal es reproducida. Este frame rate es usualmente 25-30 frames.

¿Cuántos datos que tengan valor hay en un segundo de video? Considerando el caso de un video de formato común, en el formateo intermedio común (Common Intermediate Format, CIF) operando a 30 fps, hay 360 píxeles por línea, y 288 líneas por imagen. Cada píxel toma un valor de 24 bits para el brillo y el color. De está manera, el número de bits en un solo segundo de video CIF es:

$$\text{Número de bits por segundo} = 360 \times 288 \times 24 \times 30 = 74.65 \text{ Mbps}$$

Actualmente, el dispositivo multimedia más comúnmente usado es la computadora personal. Y el dispositivo de comunicación más comúnmente usado por las PCs es el Modem, junto a las líneas de teléfono ordinarias. Utilizando un Modem a 56.6 kbps, un solo segundo de video CIF requerirá de 1296 segundos o 21.5 minutos para transmitirse. Esto significa que una computadora no puede recibir video en tiempo real. Esto también implica que una computadora deberá tener gigabits de video almacenados en memoria para recibir los datos de video, almacenarlos y reproducirlos después.

1.8 Codificación y Decodificación

Los seres humanos pueden percibir solo señales audiovisuales que son análogas. Para sistemas de comunicación digitales, estas señales análogas deben convertirse a formas digitales mediante *codificadores*, los cuáles realizan la conversión *análoga a digital (A/D)* así como la cuantización y la compresión. Muchas redes modernas de telefonía, como la ISDN, son completamente digitales en ellas se emplean transmisiones digitales y conmutación digital.

Los usuarios finales de tales sistemas de telefonía pueden ser humanos o máquinas (computadoras). Si el transmisor y el receptor son humanos, entonces las señales de origen y destino deben ser análogas, y los convertidores A/D y D/A deben de emplearse, como en la figura 1-2.

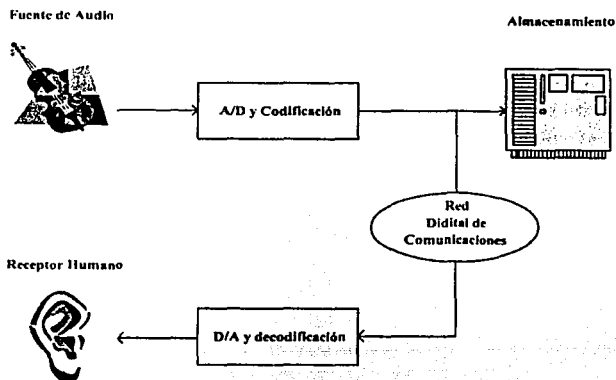


Figura 1-2 Codificación/Decodificación para una Transmisión Digital

Si, en cambio, ambos usuarios son computadoras, entonces tales convertidores no son necesarios. Aparte de la conversión D/A y A/D, el codificador/decodificador debe también realizar la función de *compresión de datos*, la cuál connota las técnicas empleadas para representar señales digitales en una forma compacta reducida.

Hay dos maneras de compresión: *lossless* (sin pérdida) y *lossy* (con pérdida). Para el primer proceso de codificación/decodificación (él *lossless*) produce una señal receptor que es idéntica a la señal enviada. Para el otro, hay una diferencia, la cuál se denota como *distorsión*. La meta de tales algoritmos es la de producir una versión altamente comprimida de la señal original, en la cuál la *distorsión* es mantenida dentro de los límites tolerados. En la mayoría de los casos, la señal comprimida es una versión a una baja tasa de bit de la señal enviada.

El uso de compresión varía de acuerdo a los diferentes medios. Los algoritmos de compresión de voz son marcadamente diferentes de las técnicas de compresión de video. La evaluación de cuales métodos de compresión son mejores que otros depende de la percepción audiovisual humana, la cuál depende sobre la percepción de la calidad de la señal. Por ejemplo en audio HI-FI, el rango de frecuencias generalmente aceptado es de 20 a 20000 Hz. Sabiendo que la mayoría de los humanos no pueden oír más allá de los 16000 Hz o debajo de 40 Hz, uno puede diseñar un algoritmo de compresión que produzca calidad de audio aceptable dentro de estos límites de frecuencia modificada. Para el video, un algoritmo de compresión diseñado debería ser uno en el cuál el número de bits por píxel pueda reducirse, digamos, de 24 a 16. O el frame rate pueda reducirse de 30 a 16 fps en escenas visuales donde no halla movimientos rápidos.

Muchas técnicas de compresión modernas son derivadas del conocimiento cercano de los mecanismos de percepción humana. Estas técnicas son referidas como *codificación perceptual*. Un ejemplo es el concepto de *distorsión notable exacta* (*Just Noticeable Distorsion, JND*) en la compresión de audio. Este es el nivel de *distorsión* en el cual hay una pequeña diferencia entre la señal original y la codificada, como lo perciben los oyentes. Para el audio HIFI la variación JND contra la frecuencia pueden ser medidos o modelados para un amplio rango de usuarios. Con el conocimiento de está curva JND, los componentes de señal más allá de la curva pueden descartarse de la compresión sin que el oyente note el efecto.

1.9 Ancho de Banda vs. Compresión

Como vimos antes, la compresión de señales es importante para las comunicaciones multimedia. Gran parte del progreso en las telecomunicaciones actuales se ha realizado tanto en la tecnología de compresión como en las redes de alta velocidad. Con la gran disponibilidad de amplios anchos de banda a bajo costo para redes locales y de grandes áreas, la necesidad de más y más señales de compresión disminuye. Sin embargo, con más y más usuarios mandando y recibiendo datos multimedia, la compresión todavía se necesita a pesar del incremento de disponibilidad de ancho de banda.

Está siendo realizado un desarrollo constante en la tecnología de compresión. Por ejemplo, en la telefonía de voz, el estándar bit rate para redes de calidad audible fue de 64 kbps en 1974. En 1984 un nuevo estándar de 32 kbps fue adoptado; y en 1992 emergió un nuevo estándar de 16 kbps. Está en proceso un nuevo estándar a 8 kbps. Tal proceso puede ser atribuido a los cada vez mejores algoritmos de compresión de voz, así como también a los cada vez más poderosos circuitos integrados para compresión de voz.

Muchos estándares internacionales de compresión usados actualmente, se aplican a redes telefónicas, audio, imagen y transmisión de video. Algunos de los estándares son listados en la tabla 1-2.

Estándar	Bit rate	Aplicación
G.721	32 kbps	Telefonía
G.728	16 kbps	Telefonía
G.722	48-64 kbps	Teleconferencia
MPEG-1 (audio)	128-384 kbps	Audio de 2 canales
MPEG-2 (audio)	320 kbps	Audio de 5 canales
JBIG	0.05-0.10 bpp	Imágenes binarias
JPEG	0.25-8.0 bpp	Imágenes fijas
MPEG-1,2 (video)	1-8 Mbps	Vides
P x 64	64-1,536 kbps	Videoconferencia
HDTV	17 Mbps	TV avanzada

Tabla 1-2 Estándares Internacionales para Telefonía, Audio y Video

Otro ejemplo de compresión es en la calidad de audio CD. Como hemos estado viendo un sonido de 20 kHz muestreado a 44.1 kHz con 16 bits por píxel abarcaría 1.412 Mbps estéreo. Mediante el uso de un *codificador de audio perceptual* (PAC), los científicos de los laboratorios Bell AT&T han demostrado la capacidad de transmitir música con calidad de CD en 64 kbps. Esto significa que un sonido con

calidad CD puede ahora enviarse sobre un canal ISDN de tasa básica. En la figura 1-3 podemos ver una tabla mostrando los límites actuales de la compresión de señales.

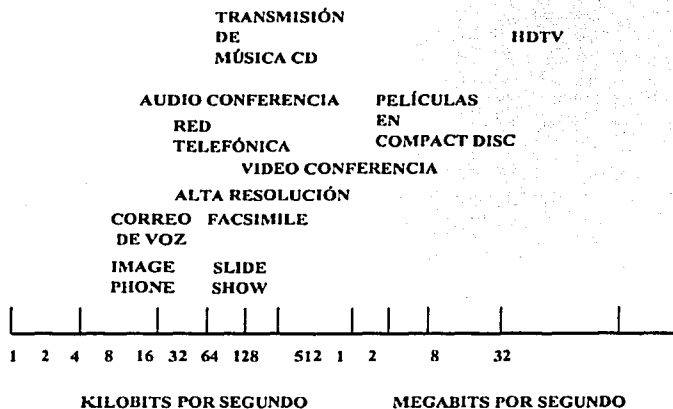


Figura 1-3 Capacidades de la compresión de señales

Una cuestión importante en la transmisión de audio y video digital sobre Internet son los anchos de banda o las tasas de las redes. Las capacidades de la red deben igualarse a la fuente multimedia que se desea transmitir. Entiéndase por fuente multimedia la información que se desea transmitir, ya sean datos, voz, imágenes fijas, audio y video.

Una caracterización de las redes y sus servicios de acuerdo a su tasa se muestra en la tala 1-3. Estas redes y sus servicios no sólo muestran una amplia variación en las tasas de transmisión disponibles, si no también el medio de transmisión físico varía dramáticamente, así como los protocolos de red. Cuando se desea transmitir fuentes multimedia comunes como las que se muestran la tabla 1-4, en las cuales se consideran sus anchos de banda, sus tasas comunes de muestreo y sus tasas de bit sin comprimir.

Al comparar los requisitos de tasas de bit sin comprimir con las tasas disponibles de red de la tabla 1-3, nos damos cuenta de la necesidad de la compresión en la fuente multimedia.

Red/Servicio	Tasa
POTS	28.8-56 kbits/s
ISDN	64-128 kbits/s
ADSL	1.544-8.448 Mbit/s (downstream) 16-640 kbit/s (upstream)
VDSL	12.96-55.2 Mbit/s
CATV	20-40 Mbit/s
OC-N/STS-N	N X 51.84 Mbit/s
Ethernet	10 Mbit/s
Fast Ethernet	100 Mbit/s
Gigabit Ethernet	1000 Mbit/s
FDDI	100 Mbit/s
802.11 (inalámbrico)	1, 2, 5.5, y 11 Mbit/s en una banda de 2.4 GHz
802.11 ^a (inalámbrico)	6-54 Mbit/s en una banda de 5 Ghz

Tabla 1-3 Redes y Servicios de red

Abreviaciones: CATV, Televisión por cable, FDDI, Interfaz para Datos Distribuidos por Fibra Óptica; OC-N/STS-N, Portador Óptico-nivel N/ Señal de Transporte Síncrona - nivel N; VDSL, Línea de Subscriber Digital de muy alta velocidad.

Fuente	Ancho de Banda (Hz)	Tasa de Muestreo	Bits por muestra	Bit Rate
Voz telefónica	200-3,400	8 kHz	12	96 kbit/s
Conversación en banda ancha	50-7,000	16 kHz	14	224 kbit/s
Audio en banda ancha (dos canales)	20-20,000	44.1 kHz/s	16 x canal	1.412 M bits/s (dos canales)
Imagen a color		512 x 512	24	6.3 Mbit/s
CCIR TV		720 x 576 x 30	24	300 Mbit/s
HDTV		1280 x 720 x 60	24	1327 Mbit/s

Tabla 1-4 Fuentes Multimedia y Bit Rates típicos sin comprimir

Abreviaciones: CCIR, Comité Internacional de Radio Comunicaciones; HDTV, Televisión de Alta Definición.

1.9.1 Eligiendo un Bit Rate apropiado

Cuando se requiere distribuir información multimedia en Internet, especialmente audio y video se hace necesario elegir un bit rate apropiado para la codificación de los archivos que van a enviarse. Además se deberá especificar el tipo de conexión del usuario designado. El tipo de conexión establece un ancho de banda máximo para tu medio de comunicación, en el caso de una transmisión en tiempo real (streaming). El ancho de banda es la cantidad de datos que pueden mandarse a través de una conexión de Internet o una red durante un periodo fijo de tiempo. Por ejemplo, un Modem a 28.8 kbps es capaz de transmitir 28,000 bits por segundo.

Sin embargo, las tasas de transmisión actuales son más bajas que los valores nominales debido a el sobre flujo de la red; el ancho de banda real de un Modem a 28.8 kbps es aproximadamente de 20 kbps, y el ancho de banda real de un modem a 56 kbps es aproximadamente de 34 kbps. Cuando se codifican archivos, el contenido no deberá exceder el bit rate de tus usuarios designados. Para asegurar una conexión estable de red, es importante no consumir todo el ancho de banda disponible para el navegador de Internet/Intranet.

La siguientes directrices de la tabla 1-5 sugieren un bit rate combinado (audio + video) con correcciones de error ligeramente debajo de la tasa de transferencia designada.

Conexión	Bit Rate total con corrección de errores	Bit Rate codificado (Audio + Video)
14.4 kbps	11,250 bps	10,000 bps
28.8 kbps	20,813 bps	18,500 bps
56 kbps	34,000 bps	32,000 bps
Single ISDN	45,000 bps	
Dual ISDN	80,000 bps	
LAN	150,000 bps	
xDSL/ cable MODEM	220,000 bps	

Tabla 1-5 Conexiones de Internet vs. Bit Rate

1.9.2 Bit Rate en Audio

El bit rate representa cuantos datos se usan para representar un segundo de audio. La selección del bit rate determina la calidad del archivo de audio. A un bit rate más alto, los sonidos de audio serán mejores. A 128 kbit/sec, el audio tiene

una calidad cercana al CD. El bit rate también determina como puede transmitirse el archivo codificado. La tabla 1-6 describe la relación entre el bit rate, la razón de compresión, la calidad de audio subjetiva, y el tipo de conexión requerido para aplicaciones streaming.

Por ejemplo si se desea codificar archivos de audio que puedan transmitirse en modems de 28.8 kbps se necesitará seleccionar un bit rate de 20 kbits/por segundo o menor.

Bit rate (kbit/s)	Tasa de compresión	Aplicación streaming	Calidad de audio subjetiva
128	11:1	Intranets	Calidad cercana a CD
96	17:1	128 kbps ISDN,	Calidad cercana a FM
64	22:1	High-speed LAN	
56	25:1	64 kbps ISDN,	Calidad FM baja
48	29:1	High-speed LAN	
32	44:1		
24	58:1	33.6 kbps modems	Calidad AM
20	70:1	28.8 kbps modems	
18	78:1		
16	88:1		

Tabla 1-6 Aplicaciones y Bit Rate

Redes Multimedia

Redes Multimedia

2.1 Aplicaciones multimedia distribuidas

En años recientes, la digitalización y la implementación de redes ha jugado un rol cada vez más creciente en la evolución para una sociedad de información distribuida. Como se nota en la figura 2-1, las aplicaciones de multimedia distribuidas son muchas y variadas. Cada una de estas aplicaciones pone requerimientos específicos para el desempeño en una red.

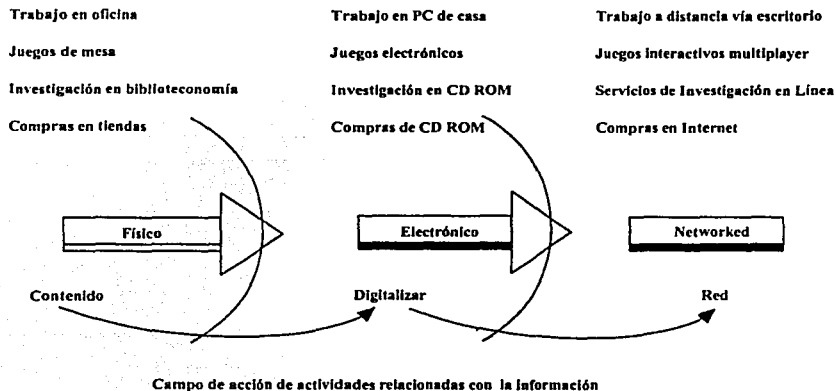


Figura 2-1 Evolución de los Servicios de Interconexión de Red

2.2 Comunicaciones Multipeer y Peer to Peer

Las comunicaciones multimedia involucran dos modos básicos: *unicast* y *multicast*. En el modo unicast, hay dos parejas de comunicadores, o pares, y el modo resultante es llamado comunicaciones par a par (*peer to peer*). El modo multicast involucra de 1 a n comunicaciones, o par a multipar (*peer to multipeer*), así como también comunicaciones uno a todos o modo de transmisión (*broadcast mode*).

Las aplicaciones unicast incluyen aplicaciones individuales cliente a servidor como por ejemplo home shopping, bancos en línea, video sobre demanda, o email multimedia.

En una aplicación de enseñanza a distancia o tele seminario, donde hay un lector y varios oyentes remotos, el modo básico es el de par a multipar. En contraste con una aplicación de tele conferencia, donde oyentes geográficamente distribuidos (pares) interactúan entre ellos vía enlaces multimedia. En una tele conferencia, cada oyente puede asumir roles de conferencista u oyente, así el modo es multipar a multipar.

Una forma especial de comunicación multipar es la de *Multyparty Interactive Multimedia (MIM)*, del cuál un ejemplo importante es *Computer Supported Collaborating Work*. En CSCW, los compañeros de trabajo distribuidos geográficamente comparten espacio de trabajo multimedia que consiste de un grupo de archivos en común, despliegues de gráficos, y un *whiteboard* distribuido. Además, ellos comparten aplicaciones tales como hojas de cálculo, editores y programas de dibujo. En una tele conferencia CSCW, los colaboradores son capaces de resolver largos diseños o problemas de ingeniería en tiempo real, mientras trabajan en diferentes lugares.

Las interacciones MIM pueden ser clasificadas como estáticas o dinámicas. Las interacciones dinámicas son aquellas en las cuáles todos los participantes tienen permitido intercambiar información a cualquier hora. En una interacción estática, solo un subgrupo prescrito de participantes tiene permitido el presentar información. Una tele conferencia multimedia, en la cuál todos los participantes son pares, es un ejemplo de interacción dinámica, CSCW es otro ejemplo. Un tercer ejemplo de interacción dinámica es el de un café virtual, en el cuál usuarios de diferentes sitios tienen permitido comunicarse informalmente a través de un ambiente electrónico compartido, muy similar a una sesión de chat de Internet. La diferencia entre CSCW y un café virtual es que el acceso a una sesión CSCW es controlada, mientras que el acceso a un café virtual no lo es.

En el modo multicast la información es pasada desde una fuente central a muchos receptores. Este es un ejemplo de interacción estática. Otro es el llamado escenario de monitoreo, en el cuál la información de varias fuentes es enviada a un solo receptor. Un tercer ejemplo es el teleteaching, en el cuál la clase es la fuente de información y todos los demás participantes son los receptores de la información. Ejemplos de estos modos diferentes de interacciones se muestran en la tabla 2-1.

Tipo	Descripción	Interacción	Flujo de Datos	Accesibilidad
CSCW	Todos pueden enviar y recibir datos	Dinámica	N a N	Controlada
Café Virtual	Todos pueden enviar y recibir datos	Dinámica	N a N	No controlada
Broadcasting	Un remitente y muchos receptores	Estática	1 a N	Controlada
Monitoreo	Un receptor y muchos remitentes	Estática	N a 1	Controlada

Tabla 2-1 Características de aplicaciones MIM

2.3 Parámetros de rendimiento de redes para multimedia

Como vimos anteriormente, hay parámetros de rendimiento de redes para comunicaciones multimedia claves. Hemos visto que la tasa de bits de una red es una característica crucial de las redes. Otros parámetros importantes son:

- *Throughput*
- *Error Rate*
- *Delay*

Cada uno de estos parámetros juegan un rol vital en la habilitación de la transmisión de señales audiovisuales a través de una red digital.

2.3.1 Throughput

El throughput de una red es la tasa de bits efectiva o ancho de banda efectivo. De está manera, nosotros definimos throughput ser el enlace físico de tasa de bit menos los varios suplementos que pertenecen a la tecnología de transmisión empleadas. Por ejemplo, en redes de alta velocidad las aplicaciones emplean tecnología ATM sobre un sistema de transmisión de fibras ópticas

SONET(Synchronous Optical NETwork), la tasa de bits provista de transmisión de una red es de 155.52 Mbps. Los principales suplementos que pertenecen son:

- Aproximadamente 3% para SONET
- Aproximadamente 9.5 % para ATM

De está manera, el máximo throughput de está red en particular es de cerca de 136 Mbps. Además de los factores de suplemento antes mencionados, otros factores también afectan el throughput. Esto incluye congestiones de red, cuellos de botella, nodos o líneas faltantes, etc.

En muchos ejemplos, el suplemento es estimado a estar implícito y el throughput es simplemente igualado a la tasa de bits del sistema. De está manera, muchas personas consideran el throughput de una tasa básica ISDN de 64 kbps, aún cuando con el throughput pueda ser de alguna manera menor debido a restricciones de suplemento.

2.3.2 Tasa de error (Error Rate)

Otro parámetro importante para las redes multimedia es el error rate. Este parámetro puede ser definido de varias maneras. Uno es el *bit error rate (BER)*, el cual es definido como la tasa del número promedio de bit corrompidos o con error con el número total de bits que son transmitidos. Otro es el *packet error rate (PER)* definido como el anterior pero en lugar de bits son paquetes. Un tercer parámetro es el *frame error rate* que se aplica a las redes ATM, definido como el número de frames con error sobre el número total de frames transmitidos.

En muchas de las redes actuales, las tasas de bit por error(los bits error rate) son muy bajas. Por ejemplo, en transmisiones de fibra óptica el rango de BERs es de 10^{-9} a 10^{-12} . En sistemas de transmisión vía satélite, los BERs son de 10^{-7} . Esto significa que en promedio hay un bit con error en un archivo de 10 000 000 de bits en un circuito digital vía satélite. Considerando que un solo frame de video consiste de varios millones de bits, tales BERs implican que puede haber aproximadamente un bit con error por cada frame en una transmisión de video digital.

Uno se podría preguntar si tales errores son importantes. En unas aplicaciones tales como la transmisión de video, un solo error en un frame podría no ser visto por el ojo humano. En otras aplicaciones tales como transferencias interbancarias de fondos electrónicos, un solo bit con error puede ser catastrófico.

2.3.3 Delay

Delay es uno de los parámetros de rendimientos de redes más importante. Puede tomar muchas formas. Nosotros discutiremos este tema desde el punto de vista *delay end-to-end*, el cuál significa el tiempo que tarda en transmitirse un bloque de datos desde el sistema que lo envía al sistema que lo recibe. Los componentes del delay end-to-end son:

- *Transit delay*, que es el parámetro físico que denota el tiempo requerido de propagación para enviar un bit de un sitio a otro, limitado por la velocidad de la luz. Este parámetro es dependiente solo sobre la distancia cruzada y su importancia cuando se usan enlaces vía satélite.
- *Transmission delay*, el cuál es definido como el tiempo requerido para transmitir un bloque de datos end-to-end. Este parámetro es dependiente por encima de la tasa de bits de la red y el procesamiento de delays en los nodos intermedios, incluyendo el enrutamiento y el almacenamiento temporal (*buffering*).
- *Network delay*, el cual está constituido de componentes del transit delay y el transmission delay.
- *Interface delay*, el cuál está definido como el delay incurrido entre el tiempo en el que el remitente este listo para empezar a enviar bloques de datos y el tiempo en que la red está lista para transmitir estos datos, éste parámetro es importante en redes de conexión orientada como las redes X.25 en las cuales un circuito end-to-end debe ser establecido antes de la transmisión de datos. Otra situación en la cuál este parámetro es tomado en cuenta es en las LANs token ring cuando la transmisión no puede empezar hasta que un token libre halla llegado.

Los componentes de delay end-to-end se muestran en la figura 2-2.

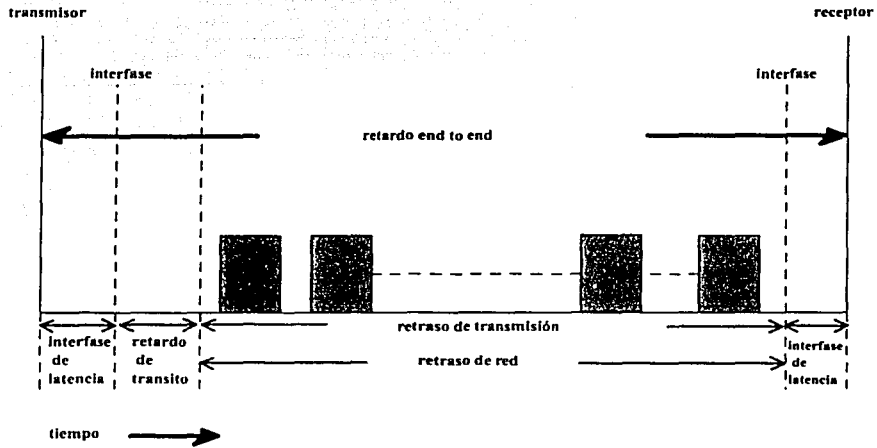


Figura 2-2 Componentes del retardo end to end

2.3.4 Round-trip Delay

El *delay round trip* es útil para redes de conexión orientada, cuando se requieren admisiones end to end. El delay round trip se define como el tiempo total requerido por el remitente para enviar un bloque de datos a través de una red y recibir un reconocimiento de que el bloque ha sido recibido correctamente. El delay round trip también juega un rol en las redes TCP funcionando al máximo en redes sin conexión IP.

Cuando las redes están muy congestionadas, este parámetro algunas veces da una mejor perspectiva del rendimiento de la red que un delay end-to-end.

2.3.5 Variación de Delay o Jitter

En las transmisiones de video digital, el flujo de datos de video y de audio son a menudo enviados separadamente. En paquetes de redes estos flujos son después divididos en bloques de datos discretos, y cada bloque es transmitido en secuencia. Si la red es capaz de transmitir todos los bloques con una latencia uniforme, entonces cada bloque llegará al destino después de un delay uniforme.

La mayoría de las redes actuales no pueden garantizar un delay uniforme a sus usuarios. Las variaciones en el delay son comunes. Aún cuando las variaciones en el delay son debidas a las imperfecciones del sistema en la red, cualquier hardware o software, o debido a las condiciones de tráfico dentro de la red; las variaciones son comúnmente referidas como *jitter*. En el diseño de una red multimedia, es importante colocar un límite superior al jitter permisible.

2.4 Características de las Fuentes de Tráfico Multimedia

La mayoría de las veces el tráfico multimedia consiste de un largo flujo de datos generados desde una fuente de audio o video digital. Aún si este flujo está partido en paquetes o frames para transportación en la red, es importante observar la integridad de los flujos por sí solos y estos a su vez asignan restricciones en los parámetros de rendimiento de la red.

¿Cómo los parámetros de rendimiento de la red afectan el tráfico multimedia en la red? Para responder esta pregunta, nosotros debemos examinar las características básicas del tráfico multimedia por sí mismo. El tráfico multimedia se compone de cinco categorías: audio, video, datos, imágenes de mapas de bits y gráficos. Las fuentes de audio y video son mayormente continuas por naturaleza. Las otras son usualmente discretas, como se definió en el capítulo anterior.

Los flujos de datos multimedia se pueden caracterizar de acuerdo a lo siguiente: *la variación throughput en el tiempo, la dependencia del tiempo y la simetría bidireccional*.

2.4.1 Variación Throughput en el Tiempo

El tráfico multimedia se puede caracterizar como una tasa de bits constantes *Constant Bit Rate (CBR)* o tasa de bits variable *Variable Bit Rate (VBR)*.

Tráfico de tasa de bit constante

Muchas aplicaciones multimedia como las aplicaciones del CDROM generan salidas a una tasa de bits constante (CBR). Para aplicaciones en tiempo real que involucra flujos de datos CBR, es importante para la red transportar estos flujos de datos a una tasa de bits constante. El no hacer esto requeriría un extenso almacenamiento temporal en cada sistema final. Es importante notar aquí que muchas redes como la ISDN transportan datos CBR de forma muy natural.

Tráfico de tasa de bits variable

El tráfico de bits variable tiene una tasa de datos la cual varía con el tiempo. Tal tráfico a menudo ocurre en despliegues o accesos. Tal tráfico intermitente se caracteriza por periodos aleatorios de relativa inactividad intercalados con los despliegues de datos. Una fuente de tráfico intermitente genera varias cantidades de datos a diferentes periodos de tiempo. Una buena medida de intermitencia es la dada por la razón de la tasa de tráfico de picos sobre la tasa de tráfico significativa en un periodo de tiempo dado.

Avances recientes en la tecnología de compresión de video han dado un aumento a flujos de tráfico de tasas de bits variables. Flujos de video VBR surgen de dos factores primordiales. Algoritmos de compresión y contenidos de escena. En escenas de movimiento lento, hay una pequeña necesidad de transmitir de frame a frame las partes estáticas de la escena tales como fondos. Sin embargo, en una escena de video en movimiento el algoritmo de compresión generará en cada frame nuevos datos representando el movimiento de los objetos. VBR puede también ser usado para conservar la capacidad de transmisión o para controlar la calidad de exposición, los flujos de video VBR son inherentemente intermitentes, pero pueden ser adaptados a redes de datos CBR.

El tráfico VBR es relativamente nuevo en las comunicaciones multimedia, por consiguiente existen hoy en día pocas formas estándar para tratar con tal tráfico.

2.4.2 Dependencia del Tiempo

En aplicaciones tales como las videoconferencias, el tráfico generado es en tiempo real, así que la latencia end to end debe mantenerse muy baja. Por ejemplo, en videoconferencias, la experiencia nos ha mostrado que el delay debe ser a lo más de 150 ms en orden para que los participantes no noten los efectos. En otras aplicaciones, tales como el e-mail multimedia, el tráfico generado no requiere que sea en tiempo real.

2.4.3 Simetría Bidireccional

Cuando dos sistemas finales son conectados por una red, el tráfico sobre está conexión es casi siempre asimétrico en su naturaleza, esto es en un canal bidireccional, el tráfico en una dirección puede ser significativamente más grande que el tráfico que va en la otra dirección. En muchos casos, el canal de avance está destinado para acarrear el flujo de tráfico, mientras que el canal reversa acarrea sólo el pequeño tráfico intermitente. Tal situación ocurre en una red por cable con aplicaciones de video sobre demanda. El solicitante hace una selección

en el canal de reversa (control) y los datos de video son enviados al solicitante en el canal de datos de avance. En contraste a esto el tráfico generado en una tele conferencia par a par puede ser generalmente considerado como simétrico si todos los participantes toman una parte activa.

2.5 Factores que afectan el rendimiento de la red

Se han definido los parámetros de rendimiento de una red que son throughput, error rate, delay, delay jitter; ahora examinaremos los factores que degradan el rendimiento de la red con respecto a estos parámetros.

2.5.1 Factores de rendimiento Throughput.

El throughput en la mayoría de las redes, sean de área local o área amplia, varían con el tiempo. Algunas veces el throughput puede cambiar rápidamente debido a repentinas fallas en nodos o líneas de la red o debido a la congestión de tráfico cuando largos flujos de datos son introducidos a la red. Factores que afectan el rendimiento general del throughput son:

- Nodos o fallas de enlace
- Congestión
- Cuellos de botella
- Capacidad del buffer
- Control de flujo

Nodos o fallas en los links.

Los nodos de red o los enlaces de transmisión cuyas operaciones son interrumpidas por cualquier motivo pueden causar congestión en otros nodos o enlaces en la inmediata vecindad del nodo (enlace aceptado). Tales fallas pueden llevar a retardos en los paquetes o pérdidas, errores en la transferencia de archivos y en algunos casos pérdida total de la conectividad. Así cuando las tasas de fallas en los nodos de red o link son usualmente bajas, las fallas ocurren y se deben tomar medidas para protegerse en contra de estos incidentes.

Congestión de red.

Cuando una red esta fuertemente cargada, la congestión puede ser debida al tráfico pesado o a los cuellos de botella, la capacidad de una red es usualmente diseñada para acomodar demandas de tráfico promedio. En ciertas horas del día o en situaciones de emergencia, la demanda para la capacidad de la red excede disponibilidad. En tales casos el throughput de la red disminuye con carga creciente debido a varios factores:

- Muchos *datagramas* de red empiezan a desechar paquetes cuando se sobrecargan los buffers de nodo.
- Los procedimientos de manejo de la red toman efecto para disminuir el tráfico de ciertos enlaces.
- Nodos cargados fuertemente se convierten en cuellos de botella.

Cuellos de botella

Otra razón para disminuir el throughput es la presencia de cuellos de botella, esto puede ser debido a fallas en nodos o links o debido a la inadecuada capacidad de los nodos o links. Un caso en este punto son los enlaces de satélite trasatlánticos que conectan redes de datos de Norteamérica a las de Europa. Muchos de estos enlaces vía satélite tienen un throughput de 128 kbps. Cuando estos links conectan dos redes de alta velocidad (como una T-1 o E-1) en lados opuestos del atlántico, representan significativos cuellos de botella. Usuarios de internet quienes tratan de conectarse al océano atlántico o pacifico experimentan a menudo estos cuellos de botella.

Capacidad de buffer

Para cada conexión end to end existe una cantidad limitada de memoria buffer en el sistema final y en las interfaces de la red (figura 2-3) los datos son temporalmente almacenados en estos buffers cuando se está enviando o recibiendo desde la red. En la transmisión de archivos largos tales como frames de video, la capacidad de buffer es a menudo inadecuada para enviar o recibir en tiempo real.

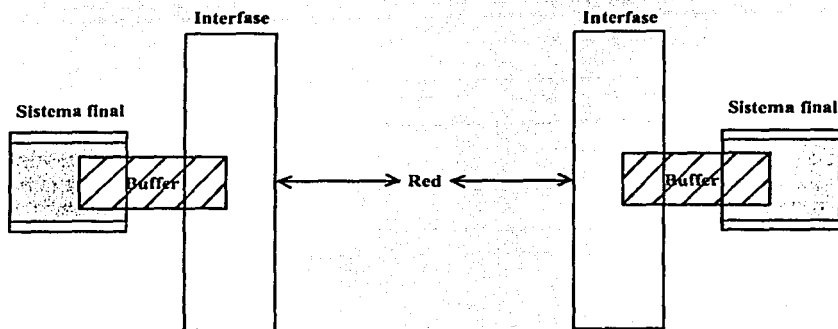


Figura 2-3 Almacenamiento temporal y conexiones end-to-end

Control de flujo

Cuando la capacidad del buffer en cualquiera de los dos puntos finales es un problema, *los protocolos de control de flujo* son a menudo invocados. El control de flujo es un protocolo end to end que pone límites en la tasa de transmisión de datos entre dos sistemas finales conectados a través de una red. Es especialmente importante si el sistema final que recibe no tiene suficiente capacidad de buffer para acomodar todos los datos que el emisor desea transmitir. El protocolo es entonces invocado para limitar o medir la tasa de datos del emisor para prevenir la pérdida de datos en el sistema final receptor. Cuando el control de flujo está en operación, se afecta el throughput end to end. Estrictamente hablando el control de flujo no debería de ser clasificado como un factor de rendimiento de la red ya que surge de las limitaciones de buffer del sistema final.

2.5.2 Cuestiones en el rendimiento de error en red

Los errores son de un gran interés en las redes *packet-switching*. Los errores de red surgen cuando:

- Los bits individuales en paquetes son invertidos o perdidos
- Los paquetes son perdidos en la transmisión
- Los paquetes son desechados o retrasados
- Los paquetes llegan sin ningún orden

Hay que notar que los paquetes perdidos son perdidos en tránsito (un error inadvertido) o desechados por un nodo intermedio (un error deliberado).

El rendimiento de error de estas redes depende de los protocolos de comunicación empleados. Algunas redes son de *conexión orientada*, en ellas antes de que el paquete o archivo sea enviado, debe establecerse primero una conexión end to end. Otras redes emplean protocolos sin conexión, lo cual es la estrategia del mejor esfuerzo, en la cual los paquetes son simplemente enviados a la red sin haber establecido primero una conexión end to end. Los esquemas sin conexión trabajan bien para mensajes cortos, y los métodos de conexión orientada son mejores para el flujo de tráfico.

Errores de Bit individuales

Con la calidad de las redes de transmisión de datos actuales, en particular, las redes de fibra óptica, los errores de bit son cada vez más raros. Sin embargo, debido a ruido en las líneas o en los interruptores de paquete, los errores de bit algunas veces ocurren. Cuando esto pasa, los códigos de detección de error empleados en la mayoría de los interruptores de paquete son capaces de hallar la presencia de un bit de error en el paquete y poder pedir la retransmisión del paquete defectuoso. La retransmisión es manejada en nodos intermedios o manejados sobre una base end to end.

La pérdida de paquetes

En una red de conexión orientada, cuando los paquetes tienen errores de bit o están perdidos o se desechan, la recepción del sistema final es usualmente capaz de detectar tal situación e informa al lado de envío el problema. El sistema final de recepción no siempre tiene una información precisa sobre que paquetes tienen errores o que paquetes han sido perdidos o se han desechado. Un acercamiento estándar al problema es para el remitente al retransmitir los paquetes más recientes al sistema de recepción final. En una red donde la corrección de error posterior (*forward error correction, FEC*) se emplea, el sistema de recepción final puede a menudo detectar y corregir errores de bit de paquetes sin acudir a una retransmisión. Sin embargo, FEC no es efectivo cuando se pierden paquetes completos o se desechan en la transmisión.

En el caso de redes sin conexión (*connectionless*), la pérdida de paquetes o la caída de paquetes son difíciles, si no imposibles, para detectar. La principal razón para paquetes que están siendo desechados o perdidos en una red de alta velocidad es el insuficiente espacio en buffer en el sistema de recepción final debido a la congestión en la red.

Paquetes fuera de orden.

Cuando un archivo largo o un flujo de datos es transmitido, normalmente los paquetes individuales en el flujo son numerados en secuencia. Este es el trabajo del receptor del sistema final al ordenar los paquetes recibidos en una secuencia numérica en la que ellos fueron enviados originalmente. Si el sistema de recepción final no tiene tal capacidad, entonces los paquetes recibidos podrían estar fuera de secuencia y de esta manera causar un error en el receptor final. Algunas veces, un paquete individual en una secuencia podría perderse, desecharse o retrasarse grandemente. En tales casos, la recepción final no es capaz de reconstruir la secuencia del paquete original, esta situación podría causar que el sistema de recepción final pidiera una retransmisión de cualquier porción o la secuencia entera de paquetes.

Problemas de rendimiento en retrasos de red.

Algunos retrasos de red son inevitables, como lo inmutable que son las leyes de la física. Si dos sistemas finales son comunicados vía una conexión satelital, entonces el retraso de tránsito de un sentido es aproximadamente de 0.25 segundos. Puesto que el satélite está a 36,000 kilómetros encima de la tierra, el tiempo en subir y bajar a la velocidad de la luz es alrededor de un cuarto de segundo. Otros retrasos son debidos a la tasa de bits de el enlace: el amplió ancho de banda del enlace hace que el retraso sea menor. Ciertos retrasos no son predecibles. La congestión en una red puede causar retrasos. Los errores de transmisión pueden conducir a retrasos. Los problemas físicos en las líneas y la interrupción de nodos pueden causar retrasos. Estos son clasificados como retrasos aleatorios y son los más difíciles al tratar con ellos.

El uso de buffers al final de cada transmisión puede a menudo dejar fuera los problemas de retraso. Por ejemplo, en un largo flujo de video, si los paquetes en el flujo son recibidos después de someterse a retrasos variados, allí debería haber muchos menos fluctuaciones (jitter) en la reproducción del flujo si fue almacenado (*buffered*) antes de la reproducción. Sin embargo, si la capacidad de los buffers es inadecuada y los flujos de video son muy largos, entonces la fluctuación probablemente no podrá ser resuelta.

Es muy deseable para la red el presentar una constante, retrasos no variables para los sistemas finales. Con un retraso constante, por ejemplo con un jitter nulo, los recursos del buffer podrían ser asignados de antemano, y la calidad de los flujos de audio y video recibido podría ser mucho más alta.

2.6 Requisitos para tráfico multimedia para redes

2.6.1 Requisitos Throughput

Los requisitos multimedia para el rendimiento throughput se discuten a continuación.

Requisitos de ancho de banda para grandes transmisiones.

Ya que el tráfico multimedia consiste de más que despliegue de datos y a menudo incluye tráfico de video y audio en tiempo real, es importante que las redes que acarrean información multimedia tengan disponible suficiente ancho de banda para transmitir y manejar este tráfico. Este requerimiento también significa que las redes deben tener la capacidad de manejar múltiples fuentes de tal información. En periodos de congestión, un insuficiente ancho de banda disponible a menudo lleva a más grandes retrasos end to end, así como también a la pérdida de paquetes.

Requerimientos de ancho de banda para grandes almacenajes.

En las redes high-throughput, es importante que el sistema final que recibe tenga suficiente capacidad de buffer para recibir el tráfico multimedia que le llega. Además, es necesario que la tasa de datos de entrada del buffer sea lo suficientemente grande para acomodar el flujo de datos entrante desde la red. Esta tasa de datos de entrada es algunas veces referida como el buffer de almacenamiento de ancho de banda. Mientras no sea una condición estrictamente de la red, el almacenaje de ancho de banda es suficientemente relevante para ser incluido en esta discusión.

Requerimientos de flujo.

Una red multimedia debe ser capaz de manejar largos flujos de tráfico tales como los que provienen de fuentes de audio y video. A lo que nos reduce esto es a que la red debe tener suficiente capacidad throughput para asegurar la disponibilidad de grandes canales de banda ancha para extensos periodos de tiempo. Por ejemplo, no es suficiente para una red el ofrecer al usuario 5 segundos de tiempo asignado (time-slot) a 1.5 Mbps si el usuario necesita enviar un flujo de tráfico de 30 Mbits. Sin embargo la red deberá presentar el requisito de flujo si puede ofrecer disponibilidad continua de 1.5 Mbps por canal al usuario. Si hay muchos

flujos en la red al mismo tiempo, la red debe tener disponible capacidad throughput igual o mayor que la tasa de bits agregada de los flujos.

2.6.2 Requisitos de confiabilidad en control de error

Es difícil cuantificar los requisitos de control de error para las redes multimedia precisamente porque las aplicaciones multimedia son hasta cierto punto tolerantes para la transmisión de errores en la red. Parte de la razón de esta tolerancia es fácil de seguir a los límites de la percepción sensorial humana. Si, por ejemplo, un error ocurre en un solo paquete en un largo flujo de video, muy a menudo el error es indetectable para el ojo humano. Una situación similar ocurre en transmisiones de audio. Sin embargo los sentidos auditivo y visual en el humano no igualmente tolerantes a tales errores. La experiencia nos ha mostrado que paquetes desechos son más notables en flujos de audio que en flujos de video. Similarmente, un paquete desecho en un flujo de texto deberá de ser mucho más notable que cualquiera de los flujos de audio o video.

Los requerimientos de control de error son también difíciles de cuantificar porque en muchos casos los requisitos de control de error y la latencia end to end son contradictorios. Las contradicciones surgen porque muchos esquemas de control de error involucran detección y retransmisión del paquete en un error o pérdida. En algunas instancias, la retransmisión debe ser llevada afuera en una base end to end que incrementa el retardo significativamente. Para transmisiones de audio o video en tiempo real el retardo es el asunto de rendimiento más importante que la tasa de error, ya que en muchos casos, es preferible olvidarse del error y simplemente trabajar con el flujo de datos recibido tal cual.

2.6.3 Requisitos de Retardo

Los datos multimedia a menudo toman la forma de múltiples flujos de datos, tales como flujos de audio y video, los cuales constituyen diferentes pero cercanamente relacionadas partes en escenas de video. En aplicaciones de tiempo real, los flujos de video y audio deben ser transmitidos a través de la red con un mínimo retardo y deben de llegar al mismo tiempo. Sin embargo, con ayuda del almacenamiento temporal, una llegada simultánea de los flujos paralelos de audio y video no es absolutamente necesaria. Cuando ambos flujos llegan esencialmente al mismo tiempo, nosotros decimos que son flujos *síncronos*.

Hay variaciones del concepto de sincronicidad que deben ser discutidas. Las siguientes definiciones son instructivas.

Asíncrona. Una conexión a red se dice que es asíncrona si no hay un límite más alto en el retardo a través de la red. Esta definición implica que la latencia puede tomar cualquier valor.

Síncrona. El término significa ocurrencia al mismo tiempo o a la misma razón con una regular o predecible relación temporal. Una conexión end to end es síncrona si dos flujos de datos cruzan la red esencialmente a la misma razón y llegan al destino del sistema final al mismo tiempo. Ya que el retardo de la red no puede ser menor que el retardo de tránsito por las limitaciones de la velocidad de la luz, una red síncrona impone sólo un estable, predecible retardo sobre y por encima del retardo de tránsito. Si el retardo adicional impuesto por la red es variable, entonces la siguiente definición es relevante.

Isócrona. Una conexión a red es isócrona si hay tanto una frontera superior y una frontera inferior a la latencia y la diferencia entre las fronteras superior e inferior es pequeña. Si la frontera inferior es de algún valor T entonces la frontera superior es $T + dT$, donde dT es pequeño. El valor dT puede ser considerado como la variación del retardo o retardo jitter, que debe ser mantenido al mínimo.

Es difícil de especificar precisamente que son los términos retardo pequeño y mínimo porque ellos realmente dependen de los requisitos de las aplicaciones subyacentes. En aplicaciones en tiempo real la latencia end to end debe mantenerse pequeña pero las variaciones de retardo deben también mantenerse a un mínimo. El problema de las variaciones de retardo es también importante cuando se trata con aplicaciones multiparty.

2.7 Calidad de Servicio

La *calidad de servicio* (QoS) es un término que indica que tan bien el rendimiento de la red está tratando con las aplicaciones multimedia. Debido a que diferentes aplicaciones imponen varios requerimientos de rendimiento sobre la red, las aplicaciones individuales han diferido expectativas de que también la red maneja sus tareas. Estas expectativas son expresadas en términos de parámetros QoS.

Hay muchas maneras de representar requerimientos QoS. Los parámetros QoS pueden ser expresados como los máximos retardos permitidos, retardos jitter, throughput, error rates, etc. Si estas aplicaciones, como conferencias en tiempo real, pueden imponer requerimientos QoS en la latencia y el throughput. Otras podrían requerir tasas de error cero pero no tener más restricciones rígidas en la latencia throughput.

Ya que los parámetros QoS pueden ser definidos explícitamente estos pueden formar una base para determinar si una red es capaz de conocer los requerimientos QoS para una aplicación dada. Si una conferencia de video *multiparty* requiere de un gran ancho de banda y un bajo retardo en una base sostenida por varias horas, y si la red usada es sujeta a congestión sobre ese periodo de tiempo, la red no podrá ser capaz de ofrecer la calidad de servicio demandada de ella por la aplicación.

En recientes sistemas multimedia experimentales, han salido nuevos conceptos QoS que pueden implementarse como servicios estándar en redes futuras.

Reservación de recursos y esquematización

Si una aplicación conocida de antemano que requiere de ciertos recursos de la red, digamos, una colocación de ancho de banda dada, puede hacer una reservación con la red para tales recursos para el periodo en cuestión. La red puede tanto negar como programar la aplicación para tal periodo y reservar los recursos solicitados.

Negociaciones de recursos

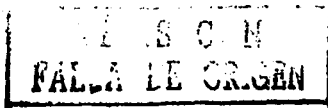
Si el administrador de la red siente que la solicitud de recursos puede sobrecargar las capacidades de la red, puede negociar con el solicitante y ofrecer parámetros QoS más bajos. Una aceptación mutua de un conjunto de parámetros QoS puede negociarse.

Control de admisión

Si las demandas QoS de la aplicación particular son tan altas que la red no puede conocerlas, la red tiene la opción de no permitir la aplicación en la red. Esta capacidad se conoce como control de admisión.

QoS garantizados

Supongamos que la red quiere atraer a un usuario en particular a su servicio: puede ofrecer al usuario potencial un servicio de calidad garantizado, en términos de ancho de banda disponible, límites superiores en retardo end to end, máximos errores permitidos, etc. Estos ofrecimientos significan que el usuario esperará un



nivel de servicios garantizados de la red. Si estas garantías son estadísticamente o dependientes absolutas sobre las negociaciones entre el usuario y la red.

Actualmente, hay muy pocas redes de datos que ofrecen garantías QoS. Una red puede apuntar al rendimiento de este servicio en bases estadísticas sobre un periodo de tiempo dado. Con información acerca de la tecnología subyacente enviada a la red y la experiencia operacional, puede obtenerse un cálculo estimado de los promedios en una red de throughput, error rates y características de latencia. Así que, una red es capaz de determinar a priori si esta puede generalmente conocer los requerimientos QoS para una aplicación en particular.

Métodos de Codificación Básicos

Métodos de Codificación Básicos

3.1 Introducción a los métodos de compresión

Con la tecnología actual, sólo la compresión hace que el almacenamiento y la transmisión de flujos de audio y video sean posibles. Cualquiera de las tasas de datos empleada en el almacenamiento y la transmisión de medios son demasiado lentas para descomprimir audio digital y datos de video o los valores requeridos para las tasas de datos son bastante altos. A menudo ambas razones son ciertas a la vez.

La compresión de datos, puede definirse como un medio de reducción del tamaño de bloques de datos al remover material redundante y sin uso. Un ejemplo de material sin uso es un periodo de silencio en una llamada telefónica, que es relativamente fácil de detectar y omitir. En contraste, esto es a menudo más difícil al extraer material redundante de flujos de audio y video digital, por ejemplo, al definir que fracción de información contenida puede ser eliminada sin la pérdida de información indispensable. Formas simples de redundancia como la repetición de datos son fáciles de describir y pueden ser codificados eficientemente, pero mucho más redundancia puede ser explotada para la compresión al analizar a fondo la facultad perceptiva humana.

Varios tipos de redundancia pueden ser explotados en los métodos de compresión:

- **Redundancia espacial.** Los valores de píxeles circundantes son fuertemente correlacionados en casi todas las imágenes naturales.
- **Redundancia en escala.** Las características importantes de una imagen como los bordes rectos y las regiones constantes son invariantes bajo el rescalamiento.

- **Redundancia en frecuencia.** En imágenes compuestas de más de una banda espectral, los valores espectrales para la misma ubicación del píxel son a menudo correlacionados y una señal de audio puede completamente enmascarar suficientemente una señal débil en su vecindad de frecuencia.
- **Redundancia temporal.** Los frames adyacentes en una secuencia de video a menudo muestran un cambio muy pequeño, y una señal de audio fuertemente en un bloque de tiempo dado puede enmascarar una distorsión baja adecuadamente en un bloque previo o futuro.
- **Redundancia estéreo.** Los métodos de codificación de audio pueden tomar ventaja de las correlaciones entre los canales estéreo.

La redundancia espacial y la redundancia en escala son típicas para los flujos de video, considerando que la redundancia estéreo puede explotar sólo la codificación de flujos de audio. Los otros tipos de redundancia pueden encontrarse tanto en flujos de audio como en video.

La mayoría de los métodos de compresión simple explotan sólo un tipo de redundancia. Las redundancias espaciales y las redundancias en frecuencia son removidas a menudo por la codificación transformada. La redundancia temporal es explotada por técnicas que codifican sólo la diferencia entre muestras o frames adyacentes, tales como el Differential Pulse Code Modulation (DPCM) y la compensación de movimiento.

Los métodos de compresión pueden ser clasificados por varias características, como lo describe la tabla 3-1. Ciertos métodos de compresión pueden a menudo ser clasificados por más de una de esas características, por ejemplo, un lossless (sin pérdida), intraframe, el método de compresión simétrica.

Muchos de los primeros métodos de compresión fueron desarrollados para la compresión de datos lossless, por ejemplo el código Huffman y el Ziv-Lempel. Pero pronto esto se haría más claro que al aplicar métodos de compresión lossless al audio y video digital son inadecuados para la transmisión de medios con un ancho de banda bajo (por ejemplo, ISDN) o para dispositivos con una baja cantidad de datos procesados (throughput), por ejemplo un CDROM. Por consiguiente, se han desarrollado métodos de compresión lossy (con pérdidas) para reducir aún más las tasas de datos. De principal interés son las técnicas que intentan extraer fuera del flujo de los datos la información que es escasamente notable al usuario, por esta razón el incremento del factor de compresión.

Características	Descripción
Lossless	Los datos originales pueden ser recuperados con precisión
Lossy	Los datos originales no pueden ser recuperados con precisión
Intraframe	Los frames son codificados independientemente
Interframe	Los frames son codificados con referencia a frames previos y/o futuros
Simétrico	El tiempo de codificación y decodificación es casi igual
Asimétrico	El tiempo de codificación excede considerablemente el tiempo de decodificación
Tiempo real	Los retrasos de codificación-decodificación no debe exceder 50 ms
Escalable	Los frames son codificados en diferentes resoluciones o en niveles de calidad

Tabla 3-1 Características de los Métodos de Compresión

La relación entre la calidad perceptible de un flujo de datos de audio o video y los requisitos del ancho de banda se describe en la figura 3-1. Mediante la aplicación de los métodos de compresión lossless (sin pérdida), los requisitos del ancho de banda pueden ser reducidos hasta cierto punto sin afectar la calidad. Al reducir el ancho de banda aún más, deben usarse las técnicas de compresión lossy (con pérdidas), por ejemplo si reducimos debajo de ciertos límites resultará una calidad baja. En años recientes, los nuevos métodos de compresión han sido desarrollados para reducir enormemente los requisitos del ancho de banda sin tener que reducir calidad para una misma extensión, es decir sólo lo que no es información o apenas es perceptible al usuario es removido del flujo de datos.

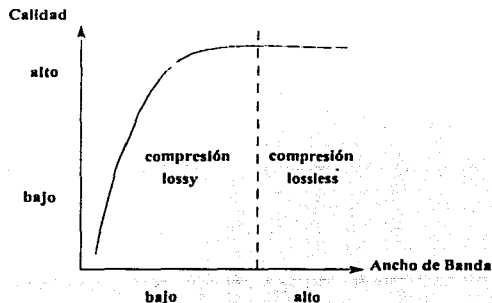


Figura 3-1 Relación entre la calidad perceptible y el ancho de banda requerido

Para su uso en sistemas multimedia, nosotros podemos distinguir entre entropía, fuente y técnicas de codificación híbridas.

Técnica de Codificación	Ejemplos
Codificación entropía	Codificación aritmética Codificación Huffman Codificación Run-length
Codificación Fuente	Modulación diferencial por códigos de pulsos Transformada coseno discreto Transformada wavelet discreta Transformada de Fourier Sistema de función iterada Predicción compensada de movimiento
Codificación híbrida	Compresión de imagen fractal H.261 H.263 JPEG MPEG video MPEG audio Codificador de audio perceptual Compresión de imagen wavelet

Tabla 3-2 Clasificación Técnicas de Codificación para sistemas Multimedia

La entropía es definida como el promedio de información contenida de los datos dados. Esto define el número mínimo de bits necesarios para representar el contenido de la información sin pérdidas de información. La codificación entropía trata de acercarse como sea posible a este límite bajo teórico. El proceso de descompresión reconstruye los datos originales completamente; por consiguiente, la codificación entropía es una técnica lossless (sin pérdida). El flujo de datos será comprimido considerando una secuencia digital simple, y las semánticas de los datos se ignoran, es decir la codificación entropía se usa para medios que no toman en cuenta sus características específicas.

La codificación source (fuente) procesa los datos originales tal que sea posible una distinción entre los datos relevantes e irrelevantes. Esto toma en cuenta la semántica de los datos. Remueve de la compresión de datos irrelevantes el flujo de datos original. En contraste a la codificación entropía, la codificación fuente es a menudo un proceso lossy (con pérdida). En el caso de las técnicas de compresión lossy (con pérdida), el flujo de datos original y el flujo de datos decodificado son similares pero no idénticos.

La codificación híbrida usa una combinación de técnicas de codificación entropía y codificación fuente. A menudo, algunas técnicas diferentes de codificación entropía y codificación fuente se agrupan juntas para formar una nueva técnica de codificación híbrida. (Ver figura 3-2). Como regla, la salida del flujo de datos producido por los pasos de la codificación fuente es usada como la entrada del flujo de datos para los pasos de la codificación entropía.

Los pasos de preparación en la figura 3-2 incluyen la conversión analógica-digital, generando una representación digital apropiada de la información.

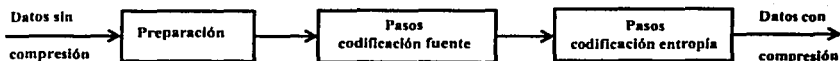


Figura 3-2 Pasos principales de la codificación de técnicas de codificación híbrida

3.2 Métodos de Codificación Básicos

Los métodos de codificación entropía, como ya se mencionó, no usan la semántica de los datos; el flujo de datos al ser procesado es considerado como un flujo de bits o bytes. Tres ejemplos de métodos de codificación entropía son mostrados a continuación.

3.2.1 Codificación Run-Lenght

La mayoría de las muestras de audio, imágenes y datos de video pueden ser comprimidas por medio de la supresión de secuencias de los mismos bytes. Estas secuencias son remplazadas por el número de ocurrencias y el patrón byte repetitivo en sí. Obviamente, el factor de compresión obtenido depende de los datos de la entrada. Usando una marca de exclamación como bandera especial para indicar la codificación run-lenght, el siguiente ejemplo muestra como los datos en el ejemplo pueden ser comprimidos al remplazar la secuencia de seis caracteres "N" con "¡6N".

Datos no comprimidos: UNNNNNNIMANNHEIM
 Codificación Run-lenght: U¡6NIMANNHEIM

Debe quedar claro que es ineficaz en este código remplazar secuencias de caracteres más cortos que cuatro porque la compresión no puede lograrse. Por ejemplo remplazando la secuencia de dos caracteres "N" con el código run-lenght

"¡2N" debería de incrementarse la longitud del código por un byte. Si en nuestro ejemplo la bandera especial aparece en los datos, ésta tiene que ser remplazada por dos marcas de exclamación ("byte stuffing").

El sencillo algoritmo presentado anteriormente es optimizado fácilmente: por ejemplo, en lugar de se secuencias de solo caracteres, secuencias más largas de diferentes caracteres pueden también remplazarse. Esta extensión requiere que cada longitud de secuencia tenga que ser codificada o una bandera al final de la secuencia tendría que usarse. Existen muchas variaciones de la codificación run-length.

Los caracteres diferentes no tienen que ser codificados con un número fijo de bits. Un ejemplo conocido es el alfabeto Morse: frecuentemente aparecen caracteres que son codificados con cadenas más cortas que las usadas por caracteres que raramente aparecen. Si la probabilidad de ocurrencia de cada carácter es dado, los caracteres pueden ser codificados con números de bits diferentes. Hay técnicas diferentes basadas en estos métodos, las más prominentes serán discutidas a continuación.

3.2.3 Codificación Huffman

Huffman desarrollo un método de compresión para determinar la codificación óptima para datos dados, por ejemplo, usando un número mínimo de bits dada la probabilidad. Aquí, la longitud en bits del los caracteres codificados diferirá. El código más corto es asignado a aquellos caracteres que aparecen más frecuentemente.

Un árbol binario ilustra el proceso para determinar el código Huffman. Los caracteres serán codificados y colocados en las hojas del árbol. Cada nodo es etiquetado con la probabilidad p de ocurrencia de uno de los caracteres en el sub-árbol. El árbol es construido por la combinación sucesiva de dos nodos con la probabilidad más baja hasta que la raíz es alcanzada. Obviamente, la raíz del árbol completo es etiquetada con la probabilidad 1. Después de la construcción del árbol, a los bordes se les asignan valores de 0 y 1. El código Huffman de cada carácter de entrada es la secuencia de etiquetas del modo en que la raíz las deja.

El siguiente ejemplo ilustra el algoritmo para un código binario. En la figura 3-3 los caracteres A, B, C y D tienen la siguiente probabilidad de ocurrencia:

$$p(A)=3/4, p(B)=1/8, p(C)=p(D)=1/16$$

La figura 3-3 describe el árbol binario junto con el proceso de reducción. El resultado de la codificación Huffman es el código siguiente que es almacenado en la llamada tabla Huffman:

$$w(A)=1, w(B)=01, w(C)=001, w(D)=000$$

La operación más costosa en determinar un código Huffman es la adición de floats (flotaciones), más específico, la suma de la probabilidad de ocurrencia en el proceso de reducción. En contraste, el decodificador tiene que realizar sólo un simple y rápido vistazo a la tabla. Sin embargo, el decodificador necesita la tabla Huffman usada por el codificador. Con dependencia en la implementación, esta tabla es cualquier parte del flujo de los datos, ya conocida por el decodificador, o tiene que calcularse mientras se ejecuta.

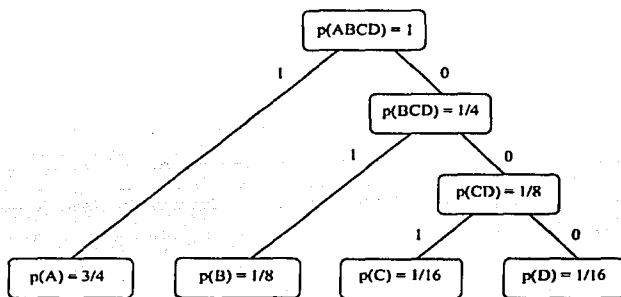


Figura 3-3 Ejemplo de codificación Huffman

Muchas variantes del código Huffman han sido desarrolladas para acelerar el proceso de codificación y decodificación, para optimizar el código Huffman resultante o para permitir una implementación eficiente. En métodos de codificación de audio y video, las tablas Huffman estándar se usan a menudo, es decir, las tablas son conocidas de antemano por el codificador y decodificador. La ventaja está en la rápida codificación por que las tablas no necesitan ser calculadas a la hora de la codificación. Una desventaja de las tablas estándar Huffman es el ligeramente mal factor de compresión por lo que las tablas no son necesariamente óptimas para los datos al ser codificados. Por lo tanto, los métodos de compresión que se ejecutan en tiempo real, a menudo usan las tablas estándar Huffman debido a que es más rápida la codificación. Si se requiere la

más alta calidad pero el tiempo de codificación no es tan importante, las tablas óptimas Huffman deberán usarse.

A menudo, no todas las entradas de caracteres tienen un código de representación en la tabla Huffman, más bien solo esos caracteres con una alta probabilidad de ocurrencia. Los otros son codificados directamente y marcados con una bandera especial. Esta técnica es útil cuando el número de caracteres de entrada diferentes es muy alto pero solo algunos de ellos tienen alta probabilidad de ocurrencia.

3.2.3 Codificación Aritmética

En teoría, la codificación aritmética es tan eficiente como la codificación Huffman, es decir la longitud de los datos codificados es mínima. En la práctica, la codificación aritmética a menudo produce ligeramente mejores resultados en la codificación de audio y video porque funciona con floats (flotaciones) en lugar de los caracteres usados en la codificación Huffman, habilitando así una aproximación cercana de la optimización matemática. Pero además el uso de floats (flotaciones) es computacionalmente más costosa y complicada una implementación, por ejemplo la representación binaria de floats (flotaciones) con valores muy bajos plantea muchos problemas.

Experimentos prácticos han mostrado que el promedio de compresión alcanzado por la codificación Huffman y la aritmética es muy similar, pero además la codificación Huffman es más rápida que la codificación aritmética. Sin embargo, muchos métodos de compresión híbridos usan una codificación Huffman optimizada como uno de los pasos de codificación entropía.

Los métodos de codificación entropía descritos anteriormente comprimen los datos mediante la eliminación de información redundante dentro de un proceso lossless (sin pérdida). Si la imagen, el audio y el video son comprimidos aún más, la facultad de percepción humana tendrá que considerarse. La codificación fuente divide los datos originales en información relevante e irrelevante, de este modo se llevan a cabo pasos consecutivos de procesamiento para eliminar los datos irrelevantes. En contraste a la codificación entropía, la codificación fuente puede ser lossy (con pérdida).

Del método de codificación fuente depende qué información se considera relevante y cuál irrelevante. Mucho más importante, las técnicas de codificación fuente explotan diferentes características de la facultad de percepción humana. Un método para separar la información relevante de la irrelevante es la codificación transformada, que transforma los datos en un modelo matemático diferente mejor preparado para el propósito de separación.

La codificación transformada más ampliamente conocida es la discrete cosine transform (la transformada discreta coseno, DCT). Otros ejemplos son Fourier transform (la transformada de Fourier) y la discrete wavelet transform (la transformada discreta wavelet, DWT). Para todas las codificaciones transformadas, una función inversa debe existir para llevar a cabo la reconstrucción de la información relevante por el decodificador. En suma, los cálculos rápidos son un importante requisito.

3.2.4 La Transformada Coseno Discreta

Todos esos requisitos antes mencionados son cumplidos por la transformada discreta coseno de una o dos dimensiones. Una DCT de una dimensión es usada en los métodos de compresión de audio; la única dimensión de interés es el tiempo. No obstante, en los métodos de compresión de imágenes las dimensiones verticales y horizontales tienen que considerarse, de esta manera requerirán de una DCT de dos dimensiones.

Para explicar como funciona DCT, discutiremos sobre la codificación de una imagen, usando una DCT de dos dimensiones. Una imagen es subdividida en bloques de muestras de 8×8 . Cada uno de estos bloques 8×8 de muestras son mapeados al dominio de la frecuencia, es decir, son representados como una composición de funciones básicas DCT con los 64 coeficientes apropiadamente escogidos, representando diferentes intensidades horizontales y verticales. Las funciones DCT básicas se muestran en la figura 3-4.

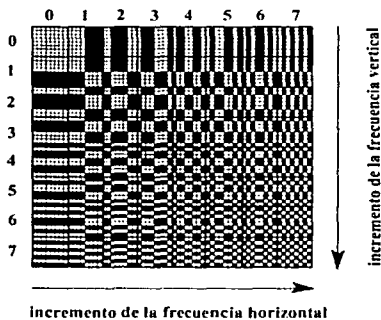


Figura 3-4 Funciones base DCT
(Cada bloque de 8×8 de muestras puede expresarse como una combinación lineal de estos 64 bloques)

El ojo humano es altamente sensitivo en niveles de baja intensidad, mientras que esta sensibilidad es grandemente reducida a niveles de alta intensidad. Una simple conclusión: una reducción del número de coeficientes de alta frecuencia DCT débilmente afecta la calidad de la imagen. En el dominio de la frecuencia, es muy fácil separar valores de alta intensidad de los niveles de baja intensidad debido a que conocemos la posición en el bloque 8 x 8 de muestras. La figura 3-5 muestra la distribución de frecuencia y las características de los coeficientes DCT.

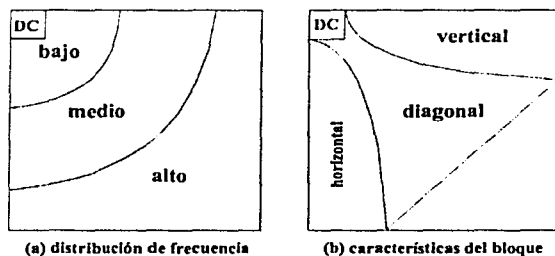


Figura 3-5 Distribución de frecuencia (a) de dos coeficientes dimensionales y características de bloque (b) su representación

Si los coeficientes transformados son ordenados en la secuencia zigzag de la figura 3-6, este nuevo ordenamiento aproximadamente corresponde con el incremento de la frecuencia y la disminución de la visibilidad. Los coeficientes con baja frecuencia (típicamente, con valores altos) son codificados primero, seguidos de la codificación de las altas frecuencias (con valores típicamente pequeños, casi cero). El coeficiente de la primera posición en la secuencia zigzag es llamado el coeficiente DC, los otros coeficientes AC.

El coeficiente DC tiene frecuencia cero en ambas direcciones la horizontal y la vertical y representan el promedio de color del bloque 8 x 8 de muestras (ver figura 3-4).

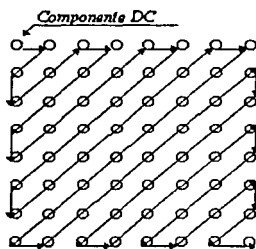


Figura 3-6 Secuencia zigzag: ordenamiento de coeficientes AC con incremento de frecuencia

En teoría, al llegar a este punto de la codificación no hay información que se haya perdido, es decir, el bloque original 8×8 de muestras puede ser reconstruido. En la práctica, la información se pierde por la precisión limitada de los cálculos. Sin embargo, DCT es usado en muchas aplicaciones sólo con números enteros y con los cálculos optimizados. Implementaciones rápidas de DCT posterior necesitan al menos de 1 multiplicación y 9 sumas en promedio para calcular un coeficiente DCT, oponiéndose a las 64 multiplicaciones y 63 sumas requeridas por la fórmula original.

Las codificaciones transformadas como DCT no comprimen los datos, La compresión se realiza por los pasos de codificación fuente después de la llamada cuantización. La cuantización usualmente se realiza al dividir cada coeficiente DCT por un entero y por redondear el resultado a un entero. La elección cuidadosa de los valores de cuantización permite a la mayoría de los coeficientes DCT con frecuencias altas ser cuantizados a un valor cero. Estos coeficientes cuantizados DCT pueden ahora ser comprimidos muy eficazmente aplicando primero una variación de la codificación run-length. De este modo se remplazan secuencias de ceros por un código run length, y después se usa una codificación Huffman modificada.

Un inconveniente del uso de métodos de codificación en bloques de muestra es la pérdida de correlación entre los límites del bloque. Es decir, a bajas tasas de bit, en factores de compresión altos, los artefactos visuales como los límites del bloque tienden a aparecer. El ejemplo de la figura 3-7 muestra ambos la imagen original y la codificada/decodificada, usando un método de codificación basado en DCT con un alto factor de compresión (1:52).

La transformada de coseno discreto es usada en muchas imágenes estandarizadas, y en métodos de compresión de audio y video. Esto muestra una superioridad en la reducción de redundancia de un amplio rango de señales tales como conversaciones, señales de TV, imágenes impresas a color, imágenes infrarrojas, etc. Otra transformada de codificación, es la transformada de Fourier, está a la vanguardia del procesamiento de señales digitales debido a que descompone una señal en sus componentes en frecuencia y tienen propiedades matemáticas simples.

En lugar de comprimir simples bytes o secuencias de bytes como en la codificación transformada, uno puede usar la codificación diferencial. Estas técnicas pueden explotar la redundancia temporal de señales de audio y video. La Differential Pulse Code Modulation (la modulación de códigos de pulso diferencial, DPCM) y la compensación de movimiento son dos importantes representantes de esta clase de métodos de codificación.



Imagen original



Imagen después de codificar/decodificar

Figura 3-7 Ejemplo de artefactos de bloque usando métodos de codificación basados en DCT

3.2.5 Differential Pulse Code Modulation

Uno de los más simples métodos de codificación fuente es el DCPM. El DCPM reduce el rango de valores de una entrada de caracteres numérica de una manera tal que los métodos de codificación sucesivos puedan lograr mejores resultados.

DPCM es a menudo aplicado en técnicas de audio. Una secuencia de muestras codificadas PCM pueden representarse suficientemente incluso si solo la primera muestra codificada es almacenada en conjunto y las muestras sucesivas son almacenadas diferentes a la precedente, es decir, son almacenados con valores predecibles.

Tomando, por ejemplo, la secuencia 10, 12, 14, 16, 18, 20. Un método de codificación run-length no puede lograr ninguna compresión, ya que todos los números son diferentes. Si la codificación DPCM es aplicada primero, por ejemplo generando la nueva secuencia 12, 2, 2, 2, 2, la codificación run-length puede comprimir a su vez la nueva secuencia significativamente, por ejemplo a 12|52.

DPCM se describió anteriormente, como un método de codificación lossless (sin pérdida) y tiene dos desventajas: Primero, ya que solo un número pequeño de bits son usados para codificar la diferencia, el error de cuantización se incrementa todo el tiempo. Segundo, La codificación de señales de audio con una oscilación de la señal muy fuerte provoca errores de cuantización altos. Las modificaciones lossy (con pérdida) de DPCM tienen que ser desarrolladas para superar estas desventajas.

3.2.6 Predicción Compensada de Movimiento

Con la predicción compensada de movimiento, pueden explotarse las redundancias temporales entre dos frames en una secuencia de video. Está técnica se usa por todos los métodos de codificación de frames internos. Las redundancias temporales pueden surgir, por ejemplo, de los movimientos de los objetos enfrente de un fondo fijo y de los movimientos de la cámara.

El concepto básico de la predicción compensada de movimiento está en ver para cierta área (bloque) un frame previo o subsecuente (frame de referencia) que iguale muy cercanamente un área del mismo tamaño en el frame actual. Es decir, si es exitoso, se encuentra un mejor bloque de igualación, después se calcula la diferencia de la señal (DPCM-codificada) entre los valores de intensidad del bloque en el frame actual y el bloque en el frame referencia. En suma, se determina el vector de movimiento, el cual representa la traslación de los bloques correspondientes en ambas direcciones x e y . Juntando, la señal diferente y el vector de movimiento representan la desviación entre el bloque referencia y el bloque predecible; sin embargo, también hay términos de *error de predicción*.

En general, tres tipos de predicción compensada pueden distinguirse:

- En la predicción compensada de movimiento unidireccional, solo un frame previo o subsecuente se usa para la predicción.

- En la predicción compensada de movimiento bidireccional, un frame de referencia anterior así como uno subsecuente se usa en la determinación del vector de movimiento para cada bloque. Solo se utiliza el vector de movimiento correspondiente al frame referencia previo o futuro que resulte con el más pequeño error de igualación. La predicción compensada de movimiento bidireccional trata con el problema de áreas cubiertas y no cubiertas.
- En la predicción compensada de movimiento interpolativo, la predicción del frame de referencia previa y la predicción del frame de referencia futura son promediados. Obviamente, dos vectores de movimiento tienen que almacenarse o transmitirse, respectivamente.

Un ejemplo de predicción compensada de movimiento unidireccional con frame de referencia previo es descrito en la figura 3-8. Ya que la predicción va del frame de referencia previo al frame actual, este tipo se llama también predicción compensada de movimiento progresivo.

Usualmente un área de búsqueda, una distancia fija en la dirección x e y, puede definirse para encontrar compatibilidad para un bloque, de este modo se limita el rango de búsqueda y la reducción de costos de extracción de información de movimiento (el proceso de la extracción de información en movimiento se llama estimación de movimiento). Los detalles de las técnicas de compensación de movimiento como métodos de búsqueda y área de búsqueda varían entre diferentes métodos de codificación interframe.

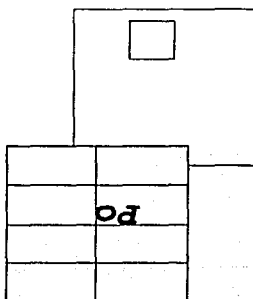


Figura 3-8 Predicción compensada de movimiento progresivo

Compresión de Video

Compresión de Video

4.1 Compresión de Video

4.1.1 Técnicas sin Estándar

Los métodos de codificación híbridos H.261, JPEG y MPEG video son actualmente puntos finales de una larga historia de desarrollo. Varios métodos para la compresión de video digital han sido desarrollados por grupos de investigación dentro de la industria y en las universidades. Los predecesores de muchos de esos métodos son Block Truncation Coding (BTC) para la compresión de imágenes monocromáticas y Color Cell Compression (CCC) para la compresión de imágenes de color.

La codificación de bloques truncados y todos estos sucesores confían en la habilidad del sistema visual humano para realizar la integración espacial sobre imágenes desplegadas o frames. Cuando se observa el área desde una cierta distancia, nuestros ojos promedian detalles finos dentro de el área pequeña(bloques) y perciben el valor promedio local de la señal. Este fenómeno puede ser explotado por la siguiente técnica: Para cada bloque de píxeles, solo dos valores de color y un patrón binario son almacenados; el patrón binario indica cuál de los dos colores es asociado con ese píxel. Algunos métodos codifican bloques que muestran una pequeña variación con un color en lugar de usar un patrón binario de dos colores.

Muchos métodos se han desarrollado para la compresión de secuencias de imágenes digitales, extensión de BTC y CCC: Extended Color Cell Compression(XCCC), desarrollada en la Universidad de Mannheim y , promovida por la industria, los métodos de codificación Digital Video Interactive(DVI), Quicktime y Software Motion Pictures(SMP), por mencionar algunos. Estos métodos difieren en la manera en que determinan la representación de un bloque



si con uno o dos colores y en como estiman y representan el patrón binario y los dos colores.

Común a todos esos métodos de compresión es el uso de tablas lookup color para llevar a cabo la reducción de datos, y la implementación de un único decodificador por software en PCs estándar o en estaciones de trabajo low-end. Por ejemplo, Indeo de Intel es la versión en software de un sistema de hardware basado en DVI. Pero la tecnología color-lookup y también la decodificabilidad única del software implican algunas desventajas en comparación a los métodos estandarizados: la calidad de las imágenes de ambas y la razón de compresión son claramente bajas. Artefactos típicos de métodos de codificación basados en CCC son descritos en la figura 4-1, mostrando la imagen comprimida Lena a 1:10 de su tamaño original.

4.1.2 JPEG

JPEG es un estándar de compresión para imágenes fijas de tonos continuos (escala de grises y de color). Se supone ser un estándar genérico para muchas aplicaciones. El JPEG estándar ofrece un modo de codificación lossless (sin pérdida) y tres lossy (con pérdida):

- Modo lossless (sin pérdida)
- Modo secuencial basado en DCT (modo línea base)
- Modo progresivo basado en DCT
- Modo jerárquico

De estos modos sólo el *modo baseline* (línea base) se presenta en más detalle porque solo este modo soportará cada aplicación JPEG. Es por mucho el método JPEG más implementado, proporcionando una decodificación estándar.

Muchos de los codificadores JPEG modernos usan sólo el método baseline. La relación entre los cuatro modos de la codificación se describe en Figura 4-2.

El modo baseline logra la compresión aplicando primero la DCT, después cuantiza, y, finalmente, codifica por entropía los coeficientes DCT correspondientes. Las unidades de codificación básica son los bloques de píxel de 8X8 en los que toda la imagen es dividida. Cada componente de imagen es codificado en un sencillo análisis de izquierda a derecha, de arriba abajo. La figura 4-3 y 4-4 describen los pasos del procesamiento de un codificador y decodificador de modo básico respectivamente.



Figura 4-1 Artefactos típicos causados por un método de codificación basado en CCC

En el modo progresivo, cada componente de imagen es codificado con múltiples escanéos en lugar de un simple análisis. El primer análisis codifica una rugosa pero reconocible versión de la imagen, que es refinada mediante análisis sucesivos hasta alcanzar la calidad de la imagen requerida. De este modo, pueden generarse imágenes de vista previa, las cuáles pueden transmitirse y presentarse muy rápido en comparación al tiempo total de la transmisión, sin la necesidad de decodificar la imagen completamente.

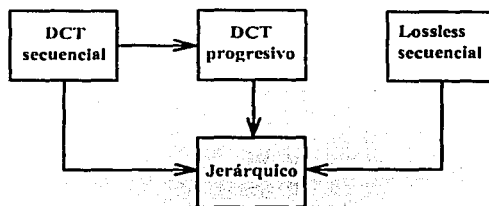


Figura 4-2 Cuatro modos de codificación JPEG

El modo lossless (sin pérdida) permite una compresión de imagen lossless (sin pérdida). En lugar de una codificación transformada combinada con cuantización como pasos de codificación fuente, un método sencillo de predicción se usa. El método lossless (sin pérdida) no puede obtener factores de compresión altos como lo hacen los modos JPEG lossy (con pérdida); no obstante, muchas aplicaciones requieren lossless (sin pérdida) para almacenamiento de imágenes, por ejemplo, fotografías en rayos X.

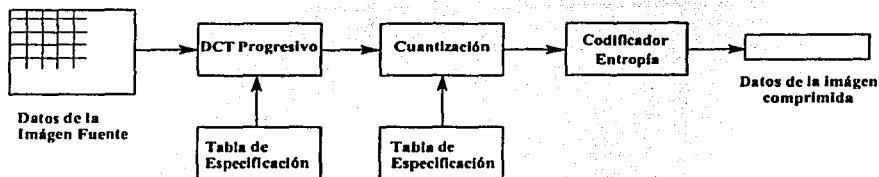


Figura 4-3 Pasos de procesamiento de un codificador modo Baseline

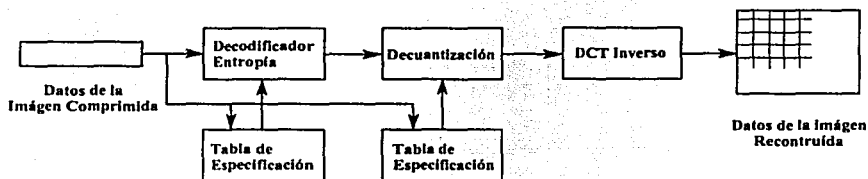


Figura 4-4 Pasos de procesamiento de un decodificador modo Baseline

El modo jerárquico codifica una imagen en múltiples resoluciones, difieren en un factor de 2 en resoluciones horizontales y/o verticales. Las versiones de resolución baja pueden accederse sin la necesidad de descomprimir primero la imagen a su resolución completa. El inconveniente para tener presente múltiples resoluciones dentro de un archivo está claro: la razón de compresión es baja en comparación a que se presentará solo una resolución. El modo jerárquico usa algoritmos definidos en los otros tres modos, como se muestra en la figura 4-2.

Preparación de la imagen en JPEG

El JPEG introduce un modelo la imagen muy general, capaz de describir la mayoría de las bien conocidas representaciones de imágenes de dos dimensiones. Cada imagen fuente debe tener un formato rectangular y puede consistir de 1 componente o plano por lo menos y 255 como máximo. Cada componente puede tener un número diferente de píxeles en el eje horizontal que en el vertical, pero todos los píxeles de todos los componentes deben codificarse con el mismo número

de bits. Cada modo define su propia precisión. Por ejemplo, los planos deben ser asignados a tres colores RGB (rojo, verde y azul), o en los tres componentes de luminancia YUV (un componente de luminancia y dos de crominancia).

Una imagen en escala de gris desearía, en la mayoría de los casos, consistir de un solo componente. Una representación de color RGB tiene tres componentes de igual resolución. En una representación de color YUV, los componentes de crominancia de una imagen son a menudo muestreados de menor calidad para reducir la cantidad de almacenamiento. Por ejemplo, en un formato 4:2:2, los componentes de crominancia tienen la mitad de la resolución horizontal del componente de luminancia; la resolución vertical es igual. En un formato 4:1:1, los componentes de crominancia tienen la mitad de la resolución horizontal y vertical del componente luminancia. La reducción de la resolución es debido al hecho de que el sistema visual humano es más sensible al componente luminancia y menos sensible a los componentes crominancia de la luz visual.

Después de la separación de la imagen fuente en sus componentes de color, cada componente es además subdividido en unidades de datos de bloques de píxel de 8x8. En muchos casos, las unidades de datos son procesadas componente por componente y entregadas, como se muestra en la figura 4-3, en este orden generado, para un procesamiento adicional al DCT progresivo. Como se muestra en la figura 4-5, para un componente, el orden de procesamiento de las unidades de información es de izquierda a derecha y de arriba hacia abajo, un componente después de otro; esto se conoce como ordenamiento de datos "sin entrelazar o intercalar". Cuando este modo sin entrelazar es usado, un decodificador JPEG debe primero decodificar la imagen completamente antes de que la imagen pueda ser desplegada. De lo contrario, el despliegue presentaría inicialmente a la imagen parcialmente decodificada en colores erróneos.

Esta desventaja puede superarse con el procesamiento entrelazado de las unidades de datos.

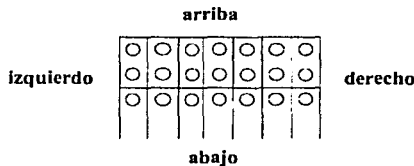


Figura 4-5 Ordenamiento de datos sin entrelazar

Procesamiento de imagen en JPEG

El modo baseline lleva a cabo la compresión aplicando primero un DCT de dos dimensiones, después cuantiza, y, finalmente, la codificación entropía a los correspondientes coeficientes DCT, como se describió en la sección 3.2.4. Correspondiente a los 64 coeficientes DCT, se usa una tabla de cuantización de 64 elementos. El coeficiente DC y cada coeficiente AC son divididos por el ingreso de la tabla de cuantización correspondiente. Después, los valores son redondeados.

Para JPEG, puede haber más de una tabla de cuantización especificada por la aplicación. Las tablas de cuantización pueden variar, dependiendo de las características de lo que se quiera desplegar, la distancia de vista, y la cantidad de ruido en la fuente. El estándar no dicta las tablas de cuantización; este sólo hace algunas recomendaciones.

Una aplicación puede influir en la cuantización mediante un factor de cuantización (factor Q). Las tablas de cuantización son multiplicadas por el factor Q. Como el factor Q se incrementa, la calidad disminuye a favor del incremento del factor de compresión. Si el factor Q es demasiado grande, tienden a aparecer artefactos visuales en la forma de los límites del bloque debido a la pérdida de correlación entre los límites del bloque. Un ejemplo para este tipo de artefactos se da en la imagen Lena derecha en la figura 3-7.

Codificación entropía en JPEG

JPEG especifica las codificaciones Huffman y aritmética como métodos para la codificación entropía. Sin embargo para el modo básico de JPEG discutido en esta sección, sólo el método de codificación Huffman se puede usar. Usualmente, las entradas al codificador entropía son algunos coeficientes no-ceros y muchos ceros-estimados. El flujo de datos reordenados zigzag de los coeficientes DCT cuantizados son procesados en dos pasos. El primer paso es una codificación run-length de los valores cero de los coeficientes AC cuantizados. Cada coeficiente AC no cero es representado por un par de símbolos, de los cuáles el primer símbolo almacena el número (run-length) de valores cero precedentes y el número de bits (tamaño) necesario para codificar su amplitud no-cero. El segundo símbolo representa la amplitud no-cero. El coeficiente DC cuantizado es codificado como la diferencia a el coeficiente DC del bloque anterior; solo se necesita el tamaño de la información en el primer símbolo. El primer símbolo es la codificación entropía usando una tabla Huffman, y el segundo símbolo es representado por un valor entero de longitud-variable.

Como se ha mencionado la codificación Huffman, es a diferencia de la codificación aritmética, no protegida por ninguna patente. Pero también a

diferencia de la codificación aritmética, la aplicación debe proveer tablas de codificación ya que el estándar JPEG no predefine cualquiera de ellas. (ver figura 4-3). El modo baseline permite el uso de diferentes tablas Huffman para los coeficientes AC y DC.

4.1.3 H.261

La Unión Internacional de Telecomunicaciones (ITU) estandariza la compresión y descompresión de video digital en 1990. Este estándar es llamado H.261. La fuerza conductora fue la necesidad de la videotelefonía y los sistemas de conferencia sobre ISDN. H.261 describe los métodos de codificación y decodificación de video para el componente de movimiento de imagen de servicios audio-visuales a una tasa de $p \times 64$ kbits/s, donde p toma valores desde 1 hasta 30, lo cuál representa el número de canales B de ISDN, teniendo cada uno una tasa de datos de 64 kbits/s. Por tanto, H.261 es también conocido como " $p \times 64$ ". Una codificación-decodificación en tiempo real con menos de 150 ms de retardo fue el enfoque de un grupo de expertos del ITU. El estándar especifica el decodificador, la sintaxis, y la semántica del flujo de bits codificado. Considerando que JPEG trata esta imagen independientemente (codificación intraframe), H.261 trata de predecir la imagen actual de una anterior (codificación interframe) para reducir la información de la imagen al ser transmitida. Los pasos de procesamiento de un codificador de video H.261 se describen en la figura 4-6.

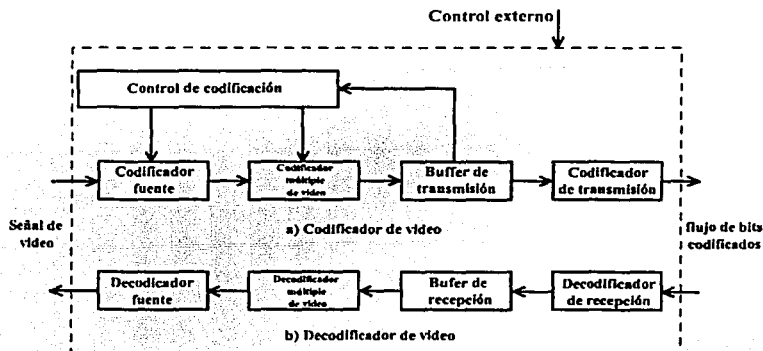


Figura 4-6 Pasos de procesamiento de un codificador de video H.261

Preparativos de la imagen en H.261

El modelo de imagen de H.261 es muy restrictivo comparado con JPEG. Una imagen en el modelo H.261 consiste de tres matrices rectangulares con 8 bits por muestra: una matriz Y luminancia y dos matrices cromáticas C_b y C_r para las componentes de cada píxel en color rojo y azul, respectivamente. Las matrices de crominancia tienen la mitad de la resolución horizontal y la mitad de la resolución vertical de la matriz de luminancia (formato 4:1:1). La ubicación de las muestras de luminancia y crominancia dentro de un frame no comprimido se describe en la figura 4-7.

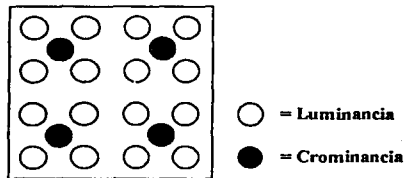


Figura 4-7 Localización de muestras de Luminancia y crominancia en H.261

Un codificador H.261 procesa imágenes fuente solamente en el formato intermedio común CIF (Common Intermediate Format) o el QCIF (Quarter CIF); ver la tabla 4-1. Todas las implementaciones H.261 deben ser capaces de codificar y decodificar QCIF; la capacidad para codificar y decodificar CIF es opcional. Los frames CIF y QCIF son divididos en una estructura de bloques jerárquicos compuesta de imágenes, de grupos de bloques (GOB), macro bloques y bloques (ver tabla 4-2).

Cada macro bloque está compuesto de cuatro bloques luminancia 8×8 y de dos bloques crominancia 8×8 . Un GOB está compuesto de macro bloques 3×11 . Una imagen QCIF tiene tres GOBs; mientras que su equivalente CIF tiene cuatro veces ese número. El procesamiento de imágenes y la codificación entropía se realizan con macro bloques.

	CIF (ancho x alto)	QCIF (ancho x alto)
Y	352 x 288	176 x 144
C_b	176 x 144	88 x 72
C_r	176 x 144	88 x 72

Tabla 4-1 Formatos de imagen fuente de H.261 (en muestras)

Elemento estructural	Descripción
Imagen	1 imagen de video (frame)
Grupo de bloques	33 macro bloques
Macro bloque	16 x 16 Y, 8 x 8 C _b , C _r
Bloque	8 x 8 pixeles (unidad de codificación para DCT)

Tabla 4-2 Estructura de bloque jerárquica

Procesamiento de imágenes en H.261

H.261 conoce dos tipos de codificación de macro bloques, llamados, intraframe e interframe. En el caso de la codificación intraframe, no se toma ventaja de la redundancia entre frames. Más allá de esto, H.261 trata de hacer uso de redundancias temporales por medio de la predicción de movimiento compensado. Si la estimación de movimiento es exitosa y la predicción de error esta debajo un cierto limite, se aplica la codificación interframe para los macro bloques.

El primer frame transmitido es siempre un frame codificado intraframe, es decir, todos los macro bloques son codificados intraframe. La imagen se codifica prudentemente en bloques. Esto es, que la imagen completa se divide en bloques de píxel 8X8 no superpuestos, en los cuáles, primero, se aplica el DCT progresivo. Después, los 64 coeficientes DCT resultantes son cuantizados y reordenados zigzag. En los preparativos para la codificación interframe, el frame codificado recientemente se decodifica otra vez dentro del codificador, usando cuantización inversa y DCT inverso. Está decodificación se realiza para obtener el mismo frame referencia exactamente del decodificador.

Para el siguiente frame a ser codificado, se usa el último frame previamente codificado y almacenado para decidir si se codifica intraframe o interframe para cada macro bloque. El algoritmo realiza una predicción unidireccional de movimiento compensada como se describe en la sección 3.2.6. El algoritmo usa solo los cuatro bloques de luminancia de cada macro bloque, especificando un área de píxel de 16X16, para encontrar una coincidencia cercana en el frame anterior para el macro bloque recientemente codificado. El área de búsqueda para el vector de movimiento es como máximo ± 15 pixeles en la dirección x e y. La unidad de compensación de movimiento de H.261 trata de detectar movimiento verificando los macro bloques. Si no encuentra una coincidencia cercana, emplea exactamente la misma codificación para el macro bloque como en la codificación intraframe. Los vectores de movimiento se codifican diferencialmente con el vector de movimiento del macro bloque a la izquierda usado de predicción. Nota que de acuerdo al estándar, el codificador no necesita determinar un vector de movimiento, es decir, una implementación sencilla H.261 considera solamente las

diferencias entre macro bloques localizados en la misma posición de los frames consecutivos.

Por consiguiente, el proceso de estimación de movimiento puede resultar de una de tres posibles decisiones para la codificación de un macro bloque:

- Intracoding, donde los valores de intensidad originales son codificación transformada
- Inter coding sin compensación de movimiento, es decir, el vector de movimiento tiene un valor cero
- Inter coding con compensación de movimiento

También opcional es un filtro entre DCT y el proceso de codificación entropía, que puede usarse para mejorar la calidad de la imagen mediante la eliminación de ruido de alta frecuencia como se necesite. Esto es especialmente útil en el caso de bajas tasas de bits.

Los bloques pueden saltarse, es decir, no son codificados, si la diferencia entre el bloque actual y el predicho es menor que un cierto umbral. Si es el mismo para los seis bloques de un macro bloque que va a ser codificado inter coding, el macro bloque completo puede saltarse. Incluso es posible saltar a tres frames entre dos frames codificados, pero por lo menos una vez cada 132 frames transmitidos, un macro bloque deberá codificarse intracoding para mitigar la propagación de error.

Diferente a JPEG y MPEG-1 Video, H.261 no hace uso de matrices de cuantización. Sólo aplica factores de cuantización a GOBs. La cuantización en H.261 es una función lineal, es decir, no distingue entre los coeficientes DCT con valores altos y valores bajos. El tamaño del paso de cuantización depende de la cantidad de datos en el buffer transformado, de este modo se impone una tasa de datos constante a la salida del codificador. Por lo tanto, la calidad de los datos de video codificados depende de los contenidos de las imágenes individuales así como del movimiento dentro de una escena de video respectiva. La cuantización de los coeficientes DC difiere de la de los coeficientes AC.

Codificación entropía en H2.261

H.261 representa un corrimiento de ceros y de valores no cero por solo un par de valores (corrimiento y amplitud). Existe una entrada de código de longitud variable para las combinaciones más probables a ocurrir. De otra manera, se usa un código (20 bits) fixed-length, que consiste de un código de escape de 6 bit, 6 bits de corrimiento, y 8 bits de amplitud. Todas las tablas Huffman usadas son

predefinidas por el estándar; estas tablas definen los vectores de movimiento, los coeficientes DCT cuantizados, etc.

4.1.4 MPEG-1 Video

MPEG Video pertenece a la familia de estándares ISO/IEC. El grupo de expertos de imágenes en movimiento (MPEG) definieron una representación de flujos de bit para audio digital sincronizado y video digital no entrelazado, comprimido para ajustarse en un ancho de banda no mayor a 1.5Mbps.

Aproximadamente 1.1Mbps son para video, 128kbps son para audio, y el restante son para el sistema MPEG. Una aplicación más grande de MPEG-1 es el almacenamiento de información audio visual en medios de almacenamiento como un CD-ROM y un DAT, que tienen la velocidad de recuperación de datos correspondiente.

MPEG-1 consiste de varias partes, de las cuales el sistema, el video y el audio son de interés principal para nosotros. El sistema MPEG-1 es el responsable del multiplexado y la sincronización de un flujo de video con uno o múltiples flujos de audio. Más allá de una simple reproducción, los flujos del sistema MPEG permiten acceso aleatorio, rebobinado y reproducción rápida progresiva.

En el desarrollo del estándar MPEG-1, fueron considerados H.261 y JPEG. Como JPEG, MPEG-1 es una codificación genérica estándar para muchas implementaciones de video digital, es decir, independientes de ciertas aplicaciones y construidas como un grupo de herramientas. Una implementación debe por tanto no realizar el estándar completo. El estándar define la sintaxis y la semántica del flujo de bits de video que deben decodificarse al final del decodificador.

El acceso aleatorio y las tasas de compresión altas son funciones que no pueden ser optimizadas ambas al mismo tiempo. El mejor acceso aleatorio puede lograrse con métodos de compresión para imágenes simples ya que cada imagen en el flujo de bits puede accederse directamente. De esta manera, la tasa de compresión es relativamente baja, ya que las redundancias temporales entre los frames sucesivos no se toman en consideración. Un ejemplo de tal método de compresión es Motion-JPEG (movimiento JPEG). Por el contrario H.261 permite cualquier acceso no aleatorio a todo o en accesos aleatorios muy restringidos, de este modo se incrementa la tasa de compresión. Para ofrecer un compromiso entre el acceso aleatorio y la tasa de compresión alta, MPEG-1 Video distingue cuatro diferentes codificaciones para imágenes descritas a continuación:

I frames (intraframe frames) son codificados sin ninguna referencia a otras imágenes. MPEG hace uso de JPEG para los frames I. Ellos son los puntos para el acceso aleatorio en flujos MPEG-1 y pueden usarse como referencia para la codificación de otras imágenes. La tasa de compresión para los frames I es el más bajo de todos los tipos de codificación.

P frames (predictively coded frames) requieren información del frame anterior I e /o el frame P para codificar y decodificar. Mediante la explotación de las redundancias temporales, la tasa de compresión que se alcanza es considerablemente más alta que la de los frames I. Los frames P pueden ser accedidos solo después de que el frame referencia I o P hallan sido decodificados.

B frames (bidirectionally predictively coded frames) requieren información del frame anterior y siguiente I e /o los frames P para codificar y decodificar. La más alta tasa de compresión se alcanza mediante el uso de estos frames debido a que, además de los métodos de codificación de los otros tipos de codificación, puede usarse una predicción bidireccional de movimiento compensado. Los frames B pueden nunca ser usados como referencia para la codificación de otras imágenes. Un ejemplo de la utilidad de los frames B se describe en la figura 4-8. El cuerpo en el frame I se oculta por la puerta cerrada; por lo tanto, la predicción unidireccional de movimiento compensado no es factible. Pero si se procede desde el siguiente frame P, por lo menos la mitad del cuerpo puede predecirse.

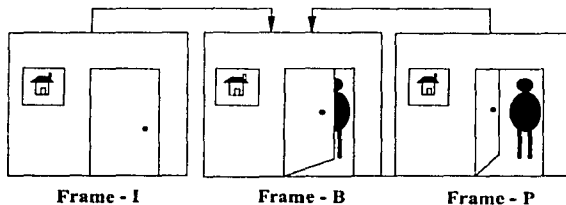


Figura 4-8 Uso de frames B en MPEG video

D frames (DC coded frame) son codificados intraframe, por lo cual los coeficientes A_c son olvidados. Pueden usarse para los modos fast-forward o el rewind (rebobinado). Los frames D nunca pueden usarse con otros tipos de imágenes, es decir, para usar el modo fast forward o rewind, ambos flujos de video (los codificados IPB y D) deben crearse y guardarse.

Los frames referencia deben transmitirse primero. Por lo tanto, el orden de la transmisión y de despliegue puede diferir, como se describe en la figura 4-9. Al

comienzo, siempre hay un frame I. El primer frame I y el primer frame P sirven como referencia para los primeros dos frames. Al mismo tiempo, el primer frame I es también la referencia para el frame P. Por consiguiente el frame I debe transmitirse primero seguido luego por el frame P y después por los frames B. Después de esto el segundo frame I debe transmitirse ya que sirve como referencia para el segundo par de frames P.

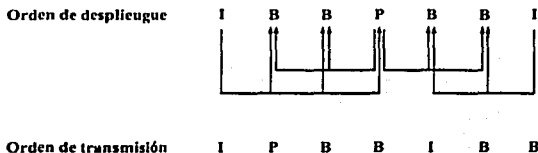


Figura 4-9 Orden de transmisión y despliegue en MPEG-1 Video

Una importante estructura de datos en la jerarquía de datos de MPEG-1 Video es el grupo de imágenes (GOP) (ver la figura 4-10). Un GOP cuenta con un número fijo de frames consecutivos que garantiza que la primera imagen de cada GOP es un frame I. Un GOP da información al codificador MPEG acerca de que imagen debe codificarse como un frame I, P o B, y cuáles frames servirán como referencias. Por ejemplo, para acceder a un frame codificado, deben codificarse a lo más un frame I, el GOP y la primera imagen del siguiente GOP. Por consiguiente, el GOP puede usarse para operaciones de edición simple en flujos de video MPEG.

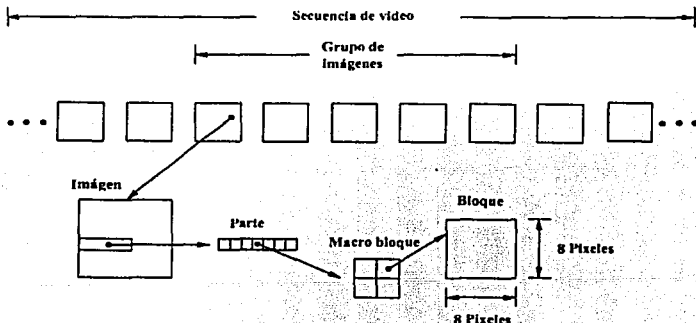


Figura 4-10 Jerarquías de datos de flujos de MPEG Video

Preparación de imágenes en MPEG-1

MPEG-1 Video emplea como H.261 el mismo formato de imagen pero permite una opción mayor del tamaño de la imagen. El componente luminancia no deberá exceder una resolución de 768 x 576. Ver la tabla 4-3

Procesamiento de la imagen en MPEG-1

Un codificador de video MPEG-1 procesa muchos diferentes formatos de imagen. Se definen dos formatos estándar de intercambio (SIF), llamados, PAL (352 x 288) y NTSC (352x 240), deducidos de dos bien conocidos estándares de televisión. Las restricciones generales son establecidas por el Constrained Parameter Set (CPS); ver la tabla 4-3.

Parámetro	Restricciones
Resolución horizontal	≤ 768 pixeles
Resolución vertical	≤ 576 líneas
Macro bloques/s	≤ 25 macro bloques/s
Frames/s	≤ 30 Hz
Rango de vector de movimiento	≤ (-64 / + 63.5) pixel
Tamaño del buffer de entrada	≤ 327.680 bit
Bit rate	≤ 1*856 Mbps

Tabla 4-3 Grupo de parámetros restringidos de MPEG-1

El primer frame en cada GOP es siempre un frame I, que se codifica atinadamente (el bloque) como una imagen codificada intraframe en H.261, es decir, con DCT, cuantización, y codificación entropía. A diferencia de H.261, el frame referencia para un frame P no es necesariamente el último frame codificado previamente. El frame referencia o los frames son decodificados otra vez dentro del codificador, usando cuantización inversa y DCT inverso. Tal como con H.261, sólo los cuatro bloques de luminancia de un macro bloque se involucran en el proceso de estimación de movimiento.

El uso del proceso de estimación de movimiento depende de la estructura de la secuencia de video, que se define por la aplicación dada GOP. El paso de estimación de movimiento se activa cuando los frames B o P aparecen en el GOP. Este es el caso más común ya que estos tipos de codificaciones logran las mejores tasas de compresión. El estándar MPEG-1 no especifica limitación alguna para

valores coordenados del vector de movimiento. Si un frame P debe codificarse, la unidad de estimación de movimiento usa el último frame I o P como frame de referencia. Si se ha encontrado una buena coincidencia entre un macro bloque en el frame actual y un macro bloque en el frame referencia, entonces el vector de movimiento y el error de predicción resultante son codificados por el macro bloque en el frame actual. De otra manera, el macro bloque es simplemente codificado intracode.

El proceso de decisión para los frames B es más complejo. Deben tomarse en consideración cuatro posibilidades: predicción progresiva, predicción retrospectiva, interpolación, o, si ninguna de estas es aplicable, la codificación intracoding de un macro bloque. Si la interpolación es aplicable, dos frames referencia deben estar disponibles: el frame I o P anterior y futuro más cercano, produciendo dos vectores de movimiento y un bloque de predicción de error. Los frames referencia para los frames P y B deben transmitirse primero.

El cuantizador MPEG usa matrices de cuantización diferentes para los bloques codificados intraframe y para cualquier otro macro bloque ya que los coeficientes DCT codificados interframe tienen propiedades estadísticas diferentes a los coeficientes DCT de codificación intraframe. El control de codificación puede ajustar la salida de la tasa de bits del codificador por medio de un factor Q. Los requisitos de la tasa de bits pueden satisfacerse de este modo, y la calidad de la imagen mejora tanto como sea posible. En caso de un error de transmisión, los macro bloques restantes de una parte son remplazados por macro bloques salteados hasta que una nueva parte comienza. A diferencia de H.261, el salteado de imágenes no se permite.

Codificación entropía en MPEG-1 Video

La codificación entropía es principalmente la misma que en H.261. Es un proceso de dos etapas en las cuáles los códigos de variables run-length y los fixed-length se aplican primero, seguidos de la codificación Huffman con las tablas predefinidas. El código de longitud variable asociado con los coeficientes DCT es un superpuesto (superset) de uno utilizado en H.261 para reducir costos cuando se implementan ambos estándares en un simple procesador.

4.1.5 MPEG-2 Video

MPEG-2 es, es al igual que MPEG-1, un método para la compresión de señales digitales audiovisuales. La parte de video de MPEG-2 permite tasas de transferencia arriba de 100Mbits/s y también soporta formatos de video entrelazado y varias características avanzadas, incluyendo las que soportan HDTV. MPEG video puede usarse para la transmisión digital de video vía

satélite, cable, y otros canales de transmisión. Esto se construirá sobre el estándar completo MPEG-1 y se desarrolló en cooperación con ISO/IEC e ITU.

Como un estándar genérico, MPEG-2 Video fue definido en términos de perfiles extensos, cada una de los cuales soporta las características necesarias para una clase importante de aplicaciones. El principal perfil de MPEG-2 fue definido para soportar transmisiones de video digital en un rango de 2 a 80 Mbps. Los parámetros del perfil principal y del perfil superior son apropiados para soportar formatos de HDTV. También, se definió un perfil escalable jerárquico para soportar aplicaciones tales como la compatibilidad terrestre de TV/HDTV, los sistemas de video packet-network, y otras aplicaciones en las que la codificación multi-nivel se requiere. Además de, la compatibilidad con estándares anteriores y los existentes (MPEG-1 y H.261).

Todas las descripciones se colocan en una matriz de 4 x 5, como se muestra en la tabla 4-4. El eje horizontal denota perfiles con un creciente número de operaciones ha soportar. El eje vertical indica niveles con los parámetros crecientes. También se declaró el límite superior para el muestreo de densidad (píxeles/línea x líneas / frames x frames/s) para cada nivel. Por ejemplo, el perfil principal en el nivel superior tiene un límite superior de 1920 píxeles / line, 1152 líneas/frames, y 60 frames/s) con una tasa de datos no menor o igual a 80Mbps.

Para codificar eficazmente señales de video entrelazado, MPEG-2 Video amplía la predicción compensada de movimiento de MPEG-1 Video y permite otras secuencias de coeficientes DCT. Además, se definen los algoritmos para la codificación escalable y jerárquica, de este modo se hace posible el uso del mismo flujo de datos para sistemas de TV normales y de HDTV. También es útil esta característica en un escenario multicast: en cada sub-árbol de una distribución multicast, sólo aquellas partes de la señal que puedan usarse por los receptores en el sub-árbol necesitaran transmitirse.

Inicialmente, MPEG-3 fue proyectado para soportar aplicaciones de HDTV. No obstante, durante su desarrollo, MPEG-2 Video probaría adecuarse cuando se adaptase a los requisitos de HDTV. Por consiguiente, MPEG-3 fue excluido.

	Perfil Simple (sin frames B, no escalable)	Perfil Principal (frames B, no escalable)	Perfil Escalable SNR (frames B, SNR escalable)	Perfil Escalable Espacial (frames B, espacial o SNR escalable)	Perfil Alto (frames B, espacial o SNR escalable)
Nivel alto (1920 x 1152 x 60)		≤ 80 Mbps			≤ 100 Mbps
Nivel 1440 alto (1440 x 1152 x 60)		≤ 60 Mbps		≤ 60 Mbps	≤ 80 Mbps
Nivel principal (720 x 576 x 30)	≤ 15 Mbps	≤ 15 Mbps	≤ 15 Mbps		≤ 20 Mbps
Nivel bajo (352 x 288 x 30)		≤ 4 Mbps	≤ 4 Mbps		

Tabla 4-4 MPEG-2 Niveles y perfiles de video con sus características más importantes

4.1.6 MPEG-4 Video

Recientemente, el grupo de expertos ISO desarrollando el estándar MPEG-4 decidió detener el desarrollo de un nuevo método de codificación de video para bajas tasas de bit. En cambio, se enfocan en proporcionar funcionalidad mejorada basada en los métodos de compresión existentes. Un ejemplo de tal funcionalidad mejorada es la codificación de objetos audiovisuales, es decir, la codificación de objetos individuales de cualquier forma en una escena de video. Considerando por ejemplo un fragmento de video que muestre a un partido de tenis. En lugar de codificar en conjunto cada imagen del fragmento de video, el fondo fijo y el jugador de tenis en el primer plano pueden codificarse independientemente con diferentes métodos o grupos de parámetros. Por el lado del audio, los objetos de audio son identificados y codificados dependiendo de sus contenidos. Ejemplos de diferentes contenidos de audio son la música, una conversación, y otros sonidos sensibles.

4.1.7 H.263

La recomendación ITU-T H.263 define un codificador para la compresión de la componente de movimiento de imágenes de servicios audiovisuales a bajas tasa de bit. Una típica aplicación es la transmisión de video sobre una conexión de modem V.34, usando 20 kbps para el video y 6.5 kbps para el audio. La recomendación se basa en H.261 y extiende este estándar en varias áreas: Primero, en lugar de dos formatos de imagen, ahora se soportan cinco. Segundo, la predicción de movimiento ha sido refinada; Tercero, el estándar ahora soporta frames B. En contraste al MPEG video, los frames B en H.263 pueden tener solo frames P como referencia. La recomendación H.263 mejora H.261 y toma en cuenta la experiencia del estándar MPEG-Video.

4.1.8 Compresión de imágenes Wavelet

La transformada de Fourier solo revela información sobre el comportamiento de la señal en el dominio de la frecuencia. Esto no se satisface bien para obtener información sobre el comportamiento de la señal en el tiempo. En los últimos años la transformada wavelet se ha vuelto una tecnología penetrante al margen en la investigación de análisis de señales y compresión. Una razón importante detrás de este desarrollo es la transformada discreta wavelet que puede decorrelacionar señales, por ejemplo, en la forma de los datos de la imagen, en casos de baja y alta frecuencia.

Aun cuando DWT es también usada en compresión de audio, la mayoría del trabajo reciente en DWT se enfoca en el análisis de señales en imágenes y en la compresión de imágenes.

¿Pero que es un wavelet? Un wavelet está definido como un conjunto de funciones base, derivadas de la misma función prototipo. La función prototipo es también conocida como la wavelet madre. Dos ejemplos de wavelet madre se muestran en la figura 4-11. Cada una de las funciones base tienen un soporte finito de diferente ancho y puede ser adaptado y transformado para satisfacer los requisitos de una aplicación. El soporte diferente permite diferentes intercambios de tiempo y resoluciones de frecuencia. Para resolver detalles con precisión de baja frecuencia, puede usarse una amplia función base para examinar una región grande de la señal, considerando que para resolver los detalles de tiempo con precisión, debe usarse una función base corta para examinar una región pequeña de la señal.

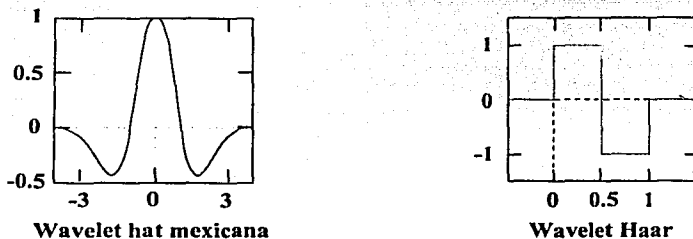


Figura 4-11 Dos wavelets madre

En contraste con los métodos de codificación basados en DCT, los cuales remueven la mayoría de la información de alta frecuencia del flujo de datos, los codificadores de transformada wavelet procesan las partes de alta y baja frecuencia de la señal independientemente. La mayor parte de la energía en las porciones de alta frecuencia de imágenes es atribuible a los bordes, ambos separan los bordes y los clusters de los bordes encontrados en regiones texturizadas. Los bordes son importantes para el reconocimiento del contenido de la imagen, pero no todos los bordes son igualmente importantes para el sistema visual humano. Dependiendo de la tasa de compresión requerida, más y más información acerca de los bordes puede removerse, de este modo se degrada la calidad de la imagen.

Además, usando la transformada wavelet, una imagen se transforma en conjunto y no en bloques de píxeles subdivididos, como los métodos de codificación basados en DCT. Por consiguiente ningún artefacto de bloque ocurre. En cambio, aún en tasas de compresión altas, los codificadores wavelet se degradan favorablemente. La imagen de la izquierda en la figura 4-12, comprimida a 1:16 de su tamaño original, muestra una muy buena calidad de imagen con ningún artefacto visual.

A un nivel de compresión de 1:256, la imagen derecha de la figura 4-12 muestra una degradación natural de la calidad de la imagen en lugar de artefactos de bloque, como con los métodos de codificación basados en DCT. (comparar con la imagen derecha de la figura 3-7).



Figura 4-12 Ejemplos de artefactos cuando se usan métodos de codificación basados en DWT

Como se menciona antes, la transformada wavelet se aplica a imágenes completas. La figura 4-13 describe un diagrama de bloques de la transformada progresiva wavelet. La imagen primero se filtra en la dimensión x, resultando una imagen paso bajas y paso altas. Puesto que el ancho de banda de ambas imágenes la paso bajas y la paso altas tienen ahora la mitad de la imagen original, ambas imágenes filtradas pueden bajarse (downsampled) por un factor de 2 sin pérdida de información. Después, ambas imágenes son filtradas y bajamuestreadas en la dimensión y, resultando 4 sub imágenes. Una de esas sub imágenes representa la señal promedio, y las tres imágenes restantes representan las características de la imagen horizontal, vertical y de la diagonal (ve también la figura 3-5).

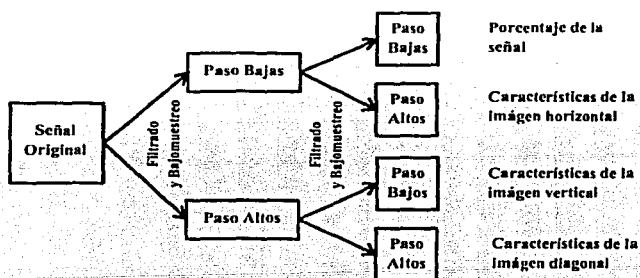


Figura 4-13 Diagrama de bloques de la transformada wavelet progresiva de dos dimensiones

La señal promedio puede transformarse recursivamente para una mejor tasa de compresión. Después de la transformada wavelet progresiva, la compresión todavía no se ha completado. En cambio, cada aplicación de la transformada wavelet progresiva incrementa el número de coeficientes, es decir, incrementa los requisitos de almacenamiento. La compresión se logra, como con DCT, mediante la cuantización y la codificación entropía de los coeficientes wavelet. Con la inversión de las operaciones de codificación, pueden reconstruirse los datos.

DCT y DWT son métodos de codificación simétricos. Debido a la mejor localización de frecuencias de DWT, tal como con DCT pueden llevarse a cabo mejores tasas de compresión con la misma calidad visual. Se han desarrollado implementaciones eficientes para DCT y DWT, con una pequeña ventaja para DCT, debida a la experiencia ganada en el campo; pero esto cambiara en el futuro. Una de las más importantes ventajas de DWT sobre DCT es la escalabilidad inherente de DWT por un factor de 2. En los métodos de codificación basados en DCT, está escalabilidad se alcanza al agregar funcionalidad, como en el modo jerárquico JPEG, resultando un factor de compresión bajo causado por el suplemento adicional.

El método de compresión de imágenes descrito en esta sección puede ampliarse para comprimir secuencias de imágenes digitales explotando las redundancias temporales en uno de dos modos. Primero, las técnicas de compresión de video estándar como la compensación de movimiento jerárquico y la codificación subbanda tridimensional pueden implementarse usando wavelets, de este modo extendiendo el DWT de dos dimensiones en la dimensión del tiempo. Segundo, en lugar de realizar la inversión completa de DWT para cada frame en una secuencia de imagen de variación lenta, uno necesita calcular solo el inverso de DWT para todos esos píxeles que han cambiado por una cantidad significativa entre los frames adyacentes en la secuencia. Experimentos han mostrado que pueden lograrse altas tasas de compresión con la primera aproximación que con la segunda. Sin embargo, la memoria y los requisitos del procesador de la primera aproximación son significativamente más altos que los de la segunda, los cuáles pueden ejecutarse muy eficazmente.

4.1.9 Compresión de Imágenes Fractal

Totalmente diferente a la codificación transformada es el método de codificación fractal de imágenes. Los codificadores de bloques fractales explotan la redundancia de imágenes a través de una transformabilidad propia en una base blockwise (modo de bloque). Ellos almacenan imágenes como mapas de reducción en las cuáles las imágenes son puntos fijos aproximados. Esta transformación lleva a una representación de una imagen como un fractal, un objeto con detalles en todas las escalas, es decir, la codificación fractal de imagen toma ventaja de la

redundancia en la escala, las técnicas descritas en esta sección pueden aplicarse a señales de audio, pero no se ha realizado mucha investigación sobre el tema.

Los codificadores transformada basados en la transformada discreta del coseno o la transformada wavelet son diseñados para tomar ventaja de estructuras muy simples de imágenes, es decir, sobre el hecho de que los valores de los píxeles que son cercanos son correlacionados. Ellos eliminan los componentes de alta frecuencia y baja frecuencia de la señal hasta el grado en que una sucesión de pasos de codificación puedan comprimir los datos. En contraste a esto, la codificación de imágenes fractales es motivada por la observación de características importantes de la imagen, tales como los bordes rectos y las regiones constantes, son invariantes bajo el rescalamiento.

Esta escala local invariante es explotada por los codificadores de bloques fractales por medio del uso de características de la imagen de una escala ordinaria para cuantizar características de escala fina. La compresión de imágenes fractales se asocia al vector de cuantización, pero esta usa un vector referencial propio del libro de códigos, traza a partir de la imagen en sí, en lugar de un libro de códigos fijo. Las imágenes no son almacenadas como un conjunto de coeficientes transformada cuantizados pero, en cambio, si como puntos fijos de mapas sobre el plano.

El libro de códigos se construye por el codificador de un promedio local y los isométricos submuestreados de los bloques más grandes de la imagen. Este libro de códigos es efectivo para codificar regiones constantes y bordes rectos debido a la escala invariante de estas características. El codificador determina un mapa de reducción de referencia propia de un plano del cuál la imagen a codificarse es aproximadamente un punto fijo. Las imágenes se almacenan al guardar los parámetros de este mapa y se decodifican al aplicar iterativamente este mapa para encontrar su punto fijo. Para almacenar eficazmente sólo los parámetros del mapa de reducción, la imagen se comprime.

En general, las transformaciones pueden inclinar, estirar, rotar, escalar y trasladar un bloque de imagen. Además, el contraste y el brillo de la transformación pueden cambiarse. El proceso de transformación de mapeo de las características de escala ordinaria a las características de escala fina se describe en la figura 4-14. En este ejemplo, el bloque original en la imagen del lado izquierdo se promedia primero y se submuestra, y después se rota. Finalmente se modifica el contraste, además de unos ajustes de compensación al bloque de imagen transformado.

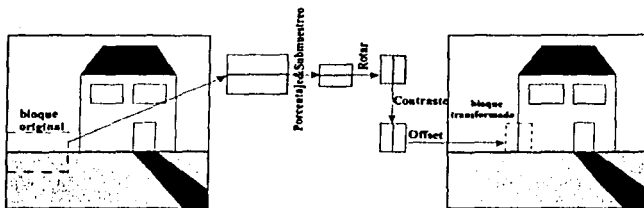


Figura 4-14 Transformación de un bloque de imagen

Es claro que los codificadores de bloque fractales son eficientes para imágenes compuestas de líneas rectas separadas y de regiones constantes, ya que éstas características son propiamente similares. En general, las imágenes deben componerse de características en una escala fina que también presente en una escala ordinaria al movimiento rígido y las transformaciones de las intensidades. Esto es la transformabilidad propia, pero esta idea también se mantiene cuando las características más complejas están presentes ya que las características complejas de la imagen tales como texturas sirven para tener características que puedan explotarse por los codificadores de fractales. Los valores de píxel en imágenes naturales se alejan de los independientes, especialmente los valores de píxel que están muy juntos.

La codificación fractal de imágenes es un método de codificación lossy (con pérdidas), ya que las imágenes naturales no son exactamente similares. La imagen decodificada es sólo una aproximación de la original, pero si las transformaciones son cuidadosamente elegidas, la diferencia entre la aproximación y la imagen original es difícil de detectar en tasas de compresión bajas. La figura 4-15 muestra en el lado izquierdo la imagen Lena comprimida 1:16 de su tamaño original y del lado derecho comprimida 1:256 de su tamaño original (compárala con la imagen Lena original en el lado izquierdo de la figura 3-7).



Figura 4-15 Ejemplos de artefactos usando un codificador de imagen fractal

Uno de los principales problemas de la compresión de la imagen fractal es la determinación de la concreta reducción de los mapas, es decir, la determinación del libro de códigos que puede usarse para codificar eficazmente una imagen. Para ese propósito, se utilizan los sistemas de función iterada (IFS). Un IFS es un conjunto finito de funciones usado por un codificador fractal para buscar similitudes propias sistemáticamente en una imagen de entrada a base de escalamiento, rotación, inclinación, o la traslación de bloques de imagen.

Compresión de Audio

Compresión de Audio

5.1 Compresión de Audio

Solamente dos dimensiones necesitan tomarse en cuenta para los métodos de codificación para flujos de audio digital, es decir, la amplitud y el tiempo. A primera vista, estás dos dimensiones reducen complejidad comparada a las tres dimensiones: ancho, altura, y tiempo del video digital. Pero el sistema auditivo humano es mucho más sensible a la calidad de degradación que el sistema visual humano, es decir, la cantidad de redundancia que puede ser removida para la compresión en relativamente pequeña. Sin embargo, aún los más avanzados métodos de codificación para audio digital tienen tasas de compresión por debajo de las del video digital. Además, la separación de información relevante de la irrelevante a menudo requiere muchos cálculos, como se mostrará a continuación.

Los métodos de compresión de audio difieren en el intercambio entre el codificador y la complejidad del decodificador, la calidad del audio comprimido, y la cantidad de compresión de datos.

5.1.1 Variaciones de Pulse Code Modulation

La tasa de datos asociados con flujos de audio codificado PCM son sustanciales. Una técnica básica de compresión de audio se emplea en la telefonía digital. Este método se basa en una transformación que en naturaleza es logarítmica; esto mapea 13 o 14 bits de valores PCM linealmente cuantizados a códigos de 8 bits. El mapeo de 13 a 8 bits se conoce como la transformación A-law, y el mapeo de 14 a 8 bits se estandariza como la transformación μ -law. A diferencia de la cuantización lineal, el espaciado de pasos logarítmicos representa muestras de audio de amplitud baja con mayor exactitud que con valores de amplitud altos. De esta manera, la razón señal a ruido de la salida transformada es más uniforme en los rangos de amplitudes de la señal de entrada. La transformación es comúnmente usada en Europa para muestreo a 8kHz en ISDN, servicio de

telefonía digital, y la transformación μ se usa en Norte América y Japón. La recomendación ITU G.711 especifica las transformaciones A-law y μ -law.

El *Adaptive Differential Pulse Code Modulation (ADPCM)* ha sido desarrollado para superar las desventajas de DPCM, como se describe en la sección 3.2.5. ADPCM es un método lossy (con pérdida) de codificación de audio que codifica las diferencias entre las señales de audio codificadas con PCM, usando solamente un pequeño número de bits (por ejemplo, 4 bits) pero puede adaptarse a las características de la señal cambiando el tamaño del paso del cuantizador, el pronosticador, o ambos. Dado tal pequeño número de bits, cualquier parte de frecuencia alta o baja de una señal puede codificarse exactamente. Sin embargo, un codificador ADPCM siempre funciona en uno de dos modos: modo de frecuencia alta o modo de frecuencia baja. En cualquiera de estos dos modos, la otra parte de la porción de frecuencia de la señal es casi ignorada.

Estas técnicas reducen la tasa de datos de audio de alta calidad de 1.4 Mbps a 32 kbps. ADPCM se estandariza como la recomendación CCITT G.721.

5.1.2 MPEG-1 Audio

Las tasas de datos alcanzadas por los métodos descritos anteriormente son todavía demasiado altas para permitir almacenamiento y transmisión de audio de alta fidelidad. Por consiguiente, el Motion Pictures Expert Group ha desarrollado dos estándares internacionales para la compresión de señales de audio: MPEG-1 Audio y MPEG-2 Audio.

Como μ -law y ADPCM, la compresión MPEG-1 Audio es lossy (con pérdida). Sin embargo, el algoritmo MPEG puede lograr claramente, compresiones perceptuales lossless (sin pérdida), como se ha probado con extensas pruebas de audición subjetivas para el desarrollo del estándar. El algoritmo MPEG explota las limitaciones perceptuales del sistema auditivo humano, es decir, el umbral auditivo y el enmascarado auditivo, para determinar que parte de una señal de audio es acústicamente irrelevante y puede removerse para la compresión.

El oído humano tiene un limitado criterio selectivo de frecuencias que varía en agudeza de al menos 100Hz para las frecuencias audibles más bajas a más de 4 kHz para las más altas. Este umbral auditivo del oído humano es descrito en la figura 5-1. De esta manera, el espectro audible puede participarse en bandas críticas que reflejan la fuerza resolutive del oído como una función de frecuencia.

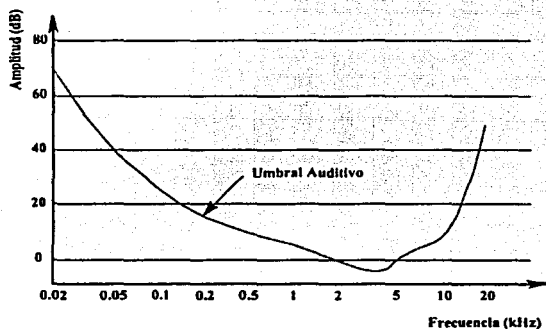


Figura 5-1 Umbral auditivo del oído humano

El enmascaramiento auditivo es una debilidad perceptual del oído que sucede cuando la presencia de una fuerte señal de audio hace una vecindad espectral de débiles señales de audio imperceptibles. El umbral para enmascarar ruido a cualquier frecuencia dada es únicamente dependiente de la actividad de la señal dentro de una banda crítica de esa frecuencia. Esta propiedad de audio de enmascarar ruido se ilustra en la figura 5-2.

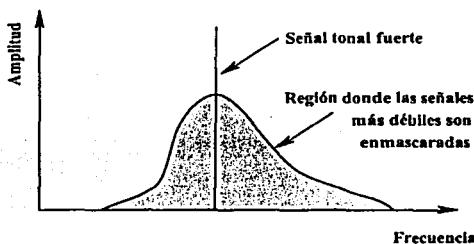


Figura 5-2 Propiedades de enmascaramiento audio/ruido del sistema auditivo humano

De este modo es esta la tarea de un codificador de audio MPEG para identificar y remover partes de la señal de audio perceptualmente irrelevantes. Un codificador de Audio MPEG trabaja dividiendo la señal de audio en subbandas de frecuencias que se aproximan a subbandas críticas y después cada subbanda se cuantiza de

acuerdo a la audibilidad del ruido de cuantización dentro de esa banda con tan pocos niveles como sea posible.

El estándar MPEG-1 Audio define tres diferentes layers (capas) de compresión; cada implementación de una capa alta debe ser capaz de decodificar señales de audio MPEG de capas bajas, es decir, un decodificador layer III (capa 3) puede también procesar flujos de bit generados por un codificador layer I o layer II. Cada layer sucesivo mejora el funcionamiento de la compresión pero a costa de mayor complejidad del codificador y el decodificador. La figura 5-3 y la figura 5-4 muestran diagramas de bloque del codificador y decodificador de audio MPEG, respectivamente.

El flujo de audio de la entrada pasa simultáneamente por un banco de filtros y por un modelo psicoacústico. El banco de filtros divide la entrada en múltiples subbandas, y el modelo psicoacústico determina la proporción de la señal a enmascar de cada subbanda. Estas proporciones son usadas por los bits o por el bloque de asignación de ruido (noise allocation block); éste determina el número de bits necesarios para hacer el ruido de cuantización a cada subbanda inaudible. El último paso formatea las muestras cuantizadas en un flujo de bits decodificable. Del lado del decodificador, el formateo se invierte, después los valores de las subbandas cuantizadas se reconstruyen y finalmente se transforman en una señal en el dominio del tiempo.

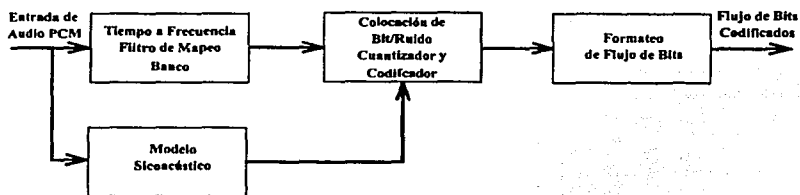


Figura 5-3 Codificador de Audio MPEG-1

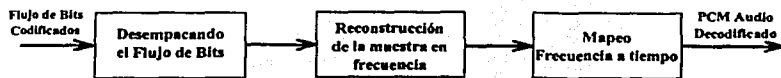


Figura 5-4 Decodificador de Audio MPEG-1

Cada layer (capa) define tasas de bit fijas para el flujo de audio codificado; sobre layer II se permiten tasas de bit variables. El valor mínimo siempre es 32 kbps. Layer I es el más simple. Este algoritmo usa un banco de filtros básico encontrado en todos los layers. Este banco de filtros divide la señal de audio en 32 bandas de frecuencias de anchura constante por medio de la transformada rápida de Fourier. Layer I usa un modelo psicoacústico relativamente simple porque su objetivo principal es la implementación fácil. Por tanto, este layer se adecua mejor a tasas de bits por encima de los 128 kbps. Por ejemplo, el Philips Digital Compact Casete (DCC) usa compresión layer I a 192 kbps por canal.

El algoritmo layer II es un perfeccionamiento directo de layer I. Tiene una tasa de bits designada de 128 kbps por canal. Su modelo psicoacústico es un poco más complicado que el de layer I. Además, usa un código más eficaz para la representación de la colocación de bits, los valores de factor de escala, y los valores cuantizados; por consiguiente, más bits codificados están disponibles para representar los valores de subbanda cuantizados. Posibles aplicaciones para este layer son el almacenamiento de secuencias de audio en CD-ROM y el track de audio de un video CD.

Layer III es el más complejo pero ofrece la mejor calidad de audio, particularmente para tasas de bit alrededor de 64 kbps por canal. Por lo tanto, este layer puede usarse para la transmisión de audio de lata calidad sobre ISDN. Además de dividir la señal de audio en las 32 subbandas descritas anteriormente, este layer compensa para algunos bancos de filtros deficiencias mediante el procesamiento de las salidas de los filtros con una transformada modificada coseno discreta, produciendo 576 supuestas líneas. Estas líneas se usan para un procesamiento posterior. Para adaptarse a los altos requisitos de este layer, el modelo psicoacústico de este layer(capa) es más complejo que el de layer I y layer II, cuya una de sus únicas características es el uso adicional de códigos Huffman para la codificación entropía.

El estándar MPEG-1 Audio define un flujo de bits que puede soportar uno o dos canales de audio: un canal simple, dos canales independientes, o una señal estéreo. Los dos canales de una señal estéreo pueden procesarse independientemente o como en *joint stereo* al aprovechar la redundancia estéreo.

El estándar MPEG-2 Audio amplía la funcionalidad de su predecesor con la codificación multicanal con arriba de 5 canales (izquierdo, derecho, central, y dos canales de surround), más un canal adicional de mejoramiento de bajas frecuencias, y/o un séptimo canal para comentarios multilinguaje. Esto también amplía la codificación estéreo y mono del estándar MPEG-1 Audio para posteriores tasas de muestreo. El estándar MPEG-2 Audio provee compatibilidad retrospectiva con el estándar MPEG-1 Audio, es decir, un decodificador de audio MPEG-2 puede procesar cualquier flujo de bits MPEG-1 Audio. Además, un

decodificador MPEG-1 Audio puede leer y procesar información estéreo de un flujo MPEG-2 Audio.

5.1.3 Codificador de Audio Perceptual

MPEG-1 Audio explota el enmascarado en el dominio de la frecuencia (también llamado simultaneous masking (enmascarado simultáneo) al separar información relevante de la irrelevante. Recientemente se han desarrollado métodos que explotan el enmascarado en el dominio del tiempo (también llamado nonsimultaneous masking (enmascarado no simultáneo) al incrementar aún más la tasa de compresión para audio digital sin degradar la calidad. Estos métodos analizan si una señal fuerte en un bloque de tiempo dado puede enmascarar una distorsión baja suficientemente en un bloque previo o posterior (enmascarado progresivo o retrospectivo).

Un ejemplo de tal método es PAC (Perceptual Audio Coder), desarrollado por los laboratorios Bell AT&T. En la prueba de audio ISO-MPEG-2, PAC demostró la mejor calidad de audio de una señal decodificada de cualquier algoritmo a 320 kbps para 5 canales de audio. Sin embargo, PAC no tiene compatibilidad retrospectiva, es decir, no puede decodificar flujos de audio MPEG-1. PAC puede proveer compresiones de audio a una tasa de codificación estéreo de 128 kbps a 160 kbps.

En los últimos años, se han alcanzado perfeccionamientos notables en la compresión de audio y video digital, que no estaban previstos en el pasado. El final de este desarrollo de ningún modo se ha alcanzado.

Flujos de Audio y Video en Internet

Flujos de Audio y Video en Internet

6.1 Flujos de audio y video en la World Wide Web

La tecnología WWW y la transferencia de datos en tiempo real como audio y video son las aplicaciones más importantes en Internet. Sin embargo, ambas tecnologías están basadas en diferentes arquitecturas de protocolo, haciendo su integración difícil. El *Hyper Text Transfer Protocol (http)* se basa en TCP. Siempre que se transfiera una página web o un archivo, HTTP establece una conexión TCP entre un pareja cliente/servidor. Dado que TCP proporciona una confiable y ordenada entrega basada en retransmisiones, es inapropiada para datos en tiempo real. Además, el intercambio de datos en tiempo real se basa típicamente en *RTP (Real Time Transport Protocol)/UDP (User Datagram Protocol)* sobre Internet.

Un simple acercamiento a la transferencia de audio y video sobre WWW es al bajar un archivo completo de audio/video de un servidor Web vía HTTP. Después de que el archivo ha sido bajado por el cliente, un programa auxiliar comienza a reproducir el sonido o el video localmente. Obviamente este proceso presenta retardos muy largos ya que los archivos de audio/video pueden ser muy largos y a menudo toma minutos bajarlos.

Para reducir este retardo, la salida de los datos en tiempo real deberá comenzar tan pronto como el primer paquete de datos haya llegado al cliente. TCP debe evitarse como protocolo de transporte para transferencia en tiempo real. El método básico para resolver este problema es al intercambiar solamente los datos de control sobre HTTP/TCP entre el cliente Web y el servidor Web para recuperar la información inicial sobre los datos de tiempo real ha transferirse. El *web browser* (navegador de Internet) en la máquina del cliente después activa el programa auxiliar de audio/video (usualmente un *plugin* del browser) inicializado con los parámetros recuperados a través de HTTP del servidor Web. Estos parámetros pueden incluir la información contenida, el tipo de codificación de los

datos de audio/video (por ejemplo, MPEG-video, PCM audio), o la dirección del servidor para ponerse en contacto para la recuperación de audio/video.

El programa auxiliar de audio/video y el servidor de audio/video ejecuta un protocolo para intercambiar la información de control requerida para reproducir y detener la transmisión de audio/video. Por ejemplo, el *Real Time Streaming Protocol (RTSP)* proporciona métodos para realizar comandos como: reproducir, adelantar, retrasar, pausa, detener y grabar, similar a la funcionalidad proporcionada por los CD players o los VCRs. RTSP puede controlar un simple flujo o varios flujos sincronizados en tiempo de medios continuos. Actúa como un control remoto de red para servidores multimedia y puede ejecutarse sobre TCP o UDP.

Los datos audio/video son recibidos por un programa cliente de audio/video, que puede ser idéntico al programa auxiliar o pueda incluso ejecutarse en una máquina cliente de audio/video distinta. El programa cliente de audio/video se habilita en cuanto lleguen los datos de salida de audio y video al cliente. Los datos de audio/video en sí son transferidos por un protocolo aparte de RTSP o HTTP, por ejemplo, RTP/UDP. El protocolo a usarse puede negociarse con el servidor de audio/video a través de RTSP.

El empleo de dos protocolos diferentes para la comunicación con el servidor audio/video nos permite la redirección de la salida de los servidores de audio/video a otra dirección destino diferente de la máquina en la que el programa auxiliar de audio/video se ejecute. Por ejemplo, a través de HTTP un cliente puede recobrar de un servidor Web información de audio/video como el tipo y contenido de un archivo de audio/video almacenado en cierto servidor de audio/video. El programa auxiliar de audio/video entonces da instrucciones al servidor de audio/video para mandar los datos de audio/video a una dirección multicast o a una dirección unicast de otra máquina cliente de audio/video. (Ver Figura 6-1)

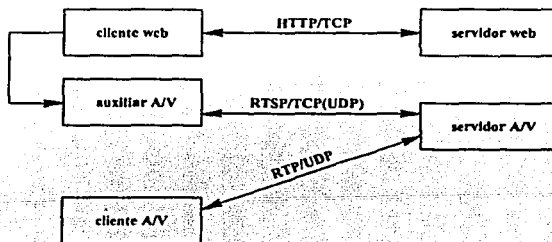


Figura 6-1 Transferencia de datos en tiempo real a través de Internet

Por ejemplo, un profesor puede preguntar a un servidor Web por videos interesantes de aprendizaje sobre cierto tema. El servidor Web regresa varias direcciones de servidores de video correspondientes con los nombres o identificadores del material de aprendizaje, como archivos de audio/video, para el profesor. El programa auxiliar del profesor después contacta al servidor de audio/video y da instrucciones para transmitir el archivo de audio/video a la dirección de grupo multicast, especificando el conjunto de aprendices. RTSP puede también usarse para dar instrucciones a un servidor de audio/video para grabar sesiones de audio/video como una lectura. En este caso, el servidor de audio/video se activa y da instrucciones para grabar los datos de audio/video transmitidos de un cliente de audio/video a una dirección multicast.

6.2 Streaming video en Internet

El video ha explotado recientemente en Internet del mismo modo en que las conexiones rápidas de Internet se han vuelto más comunes. Esto ya no es inusual en sitios Web de negocios y en páginas privadas que incluyan secciones de audio y video para ayudar a comunicar lo que las palabras y gráficos hace algunos años comunicaban. Inicialmente el único método de entrega a un usuario de Internet era hacer pequeños archivos de video disponibles, los cuales podían ser bajados a un PC o Mac local, normalmente un proceso que consumía mucho tiempo. Una vez que el archivo había sido completamente bajado, un reproductor estándar podía usarse para abrir y ver los archivos resultantes. Otro método, la descarga progresiva (algunas veces conocido como *http streaming*), ayudaba a reducir el tiempo de espera permitiendo a los usuarios empezar a reproducir el video antes de que el archivo entero fuera salvado en un archivo temporal en la computadora. Dentro del último año el video "*true streaming*" se ha vuelto muy popular, permitiendo a los usuarios reproducir video desde un *servidor streaming*.

6.2.1 Definiciones básicas

Streaming (o web streaming) es la capacidad de enviar datos directamente de un servidor a tu computadora y empezar a usar esos datos en cuanto son recibidos. En el caso del video streaming, puedes empezar a reproducir el video sin tener que esperar la descarga del archivo completo del servidor a tu computadora. Esto resulta en relativamente pequeñas esperas de solo algunos segundos antes de que el archivo comience a reproducirse, y no requiere que la persona que este viendo el video almacene el archivo completo de video en su disco duro antes de que pueda ver el video.

Live Web Broadcasting (o live Webcasting)

Es la capacidad de transmitir y reproducir desde una computadora, mandando la transmisión a un servidor streaming en uno de los dos formatos soportados web streaming. Otras personas que accesan a la misma dirección del servidor pueden ver el video transmitido al mismo tiempo. Si el usuario final está usando streaming o webcasting en vivo, su *ISP (Internet Service Provider)* debe soportar streaming. Si tu ISP no soporta streaming, podrás tener hospedados tus clips streaming en un proveedor de servicio dedicado a streaming.

6.2.2 Definiciones de servidor

Hay dos tipos de servidores que son usados con video streaming: un streaming server y un *regular web server*. El envío de un video desde un streaming server involucra la transferencia de paquetes de datos que intente igualar la velocidad de conexión del usuario. Inicialmente el servidor de streaming requiere que la computadora del usuario determine la velocidad más rápida de conexión para esa máquina. El video puede ser mandado en paquetes de datos que llegarán a tiempo para permitir que el video se reproduzca continuamente sin exceder la velocidad de conexión del usuario. Aún cuando se use un servidor streaming algunos datos necesitarán almacenarse antes que la reproducción inicie. Esto permite que el video se reproduzca continuamente aún cuando haya una congestión de red y la disponibilidad de datos disminuya.

Hay tres formas en que el video puede enviarse al usuario desde un servidor streaming:

- ***Unicast***: Un protocolo punto a punto en el cuál una copia del video será enviada a cada usuario, cada vez que un nuevo usuario empiece a ver el video un nuevo flujo de datos se genera en el servidor y se envía al usuario. Este es el método que se usa cuando un archivo de video es almacenado en un servidor streaming.
- ***Multicast***: Un protocolo punto a multipunto en el cuál un flujo de datos es enviado y muchos usuarios pueden ver el video. Este método se usa cuando una transmisión en vivo programada está siendo enviada a internet.
- ***Multicast Reflected***: En el cuál un servidor intermedio recibe datos desde una transmisión unicast o multicast y entonces puede retransmitir a múltiples usuarios.

El envío de video de un *regular web server* usando http streaming es un proceso normal de transferencia de archivos. El archivo se copia a la computadora del

usuario a la mayor velocidad posible, y comienza a reproducirse tan pronto como un conjunto de datos se haya almacenado en el disco duro del usuario. El mismo archivo completo se envía a todos los usuarios. Una vez que el archivo ha sido recibido por el usuario, son las opciones del reproductor las que determinan la tasa de bits usada cuando el archivo se reproduce. Esto es porque el archivo está siendo reproducido desde el disco duro del usuario, y no desde el servidor.

6.2.3 Formatos de archivos de video para Internet

Los formatos de archivo más populares de streaming video usados en Internet actualmente son: Real Media, Microsoft ASF/WMA y Quicktime. Quicktime, MPEG-1 y AVI también son archivos de video disponibles muy populares que pueden ser bajados a tu máquina local antes de reproducirse. En los comienzos del video en Internet, estos formatos de archivo que se pueden bajar fueron usados para proporcionar bajas resoluciones de video en las computadoras. Estas capacidades de formato han evolucionado y pueden usarse para proporcionar video de resolución completa cuando se maneja video digital en el disco duro de una computadora. Sin embargo, la baja resolución de video aún es típicamente usada en Internet debido a las limitaciones de la capacidad de datos en las conexiones típicas de Internet.

La nueva generación de formatos de archivo de video de Real Networks (RM), Microsoft (WMV,WMA) y Apple (Quicktime MOV con "Hinting") proporcionan lo que se conoce como "archivos de formato true streaming". Estos archivos de video pueden ponerse en servidores de streaming especializados (como el Real Server) y pueden reproducirse usando "true streaming". Hay que notar que los mismos archivos streaming pueden también reproducirse desde un disco duro, un disco compacto o un servidor HTML usando descargas progresivas (o HTML streaming).

Cuando se hace un archivo Real Media de Real Networks (RM) o un archivo de Windows Media Format(WMV), no se necesita habilitar ninguna configuración especial para que el archiv pueda transmitirse a través de internet. Cuando se haces un archivo Apple Quicktime se debe hacer con la función "hinting" habilitada si se quiere que el video se transmita a través de Internet.

Cuando se hace un archivo de video streaming una variedad de opciones te permiten hacer el video streaming accesible para varias conexiones de Internet a varias velocidades. Tecnologías como la SureStream (Real Networks), Intelligent Streaming (Microsoft) o hinting (Apple) permiten a los archivos streaming ajustarse automáticamente a la adecuada tasa de datos durante la reproducción actual. Estos archivos streaming de múltiples velocidades (o múltiple bit rate files) pueden bajarse de servidores web normales, pero ellos pueden hacer sólo uso

de las capacidades de tasa de bit múltiple cuando son enviados desde un servidor streaming.

6.2.4 Configurando una computadora para ver video streaming

Una manera fácil para comenzar a ver video streaming desde una computadora es tan simple como bajar uno de los reproductores gratuitos de Real Networks, Microsoft o Apple. Estos reproductores también cuentan con plugins para el navegador de Internet. Aparte del reproductor adecuado, la persona que vea el video no necesita ningún otro software instalado. Una de las ventajas inmediatas de esta nueva tecnología streaming es que nos permite buscar en la red relativamente rápido archivos de video como eventos deportivos, noticias, videos musicales y avances de películas. No se necesita esperar hasta que el archivo completo se baje a la computadora para ver un fragmento como se usaba en el viejo método. Sin embargo, la calidad de reproducción está a merced de la capacidad de la conexión de Internet y que tan ocupados estén los servidores y los sitios de Internet.

6.3 Trabajo en Internet del Audio

El audio vía Internet tiene un vasto rango de aplicaciones. Fragmentos de audio como mensajes, música de fondo, sonidos en páginas Web, estaciones de radio y streaming video son solo algunas de las aplicaciones en las que se involucra el audio. Uno de los roles más importantes del audio a través de Internet es su reproducción continua en transmisiones de radio, en las cuales el usuario pueda sintonizar distintas estaciones de radio de cualquier parte del mundo.

Sin embargo el audio vía Internet tiene varios obstáculos para llevar a cabo una buena transmisión de audio con calidad:

Calidad de Audio. Para escuchar una transmisión en tiempo real, la fidelidad del audio vía Internet es equivalente a una transmisión de radio AM (Amplitud Modulada), adecuada para noticias y otra programación que involucre la voz, pero no es adecuada para alguien que escucha música de mayor calidad.

Rentabilidad. Internet es una red donde se comparte información y la calidad del servicio varía porque nunca fue diseñada para transmitir audio en tiempo real.

Costo. Dependerá de cuantos usuarios quieran conectarse a la misma transmisión de audio online, ya que cada usuario se conecta a un servidor Webcasting individualmente. Esto significa que el *Webcaster* (proveedor de servicio) tiene que

pagar para todos la capacidad del servidor para transmitir a usuarios simultáneos y la conectividad a todos los usuarios en conexiones separadas.

Conexión a Internet	Ancho de Banda	Usuarios	Costo por Mes (dlla)
ISDN Basic Rate	128 Kbps	8	\$250
T1	1.5 Mbps	90	\$2,000
Ethernet LAN	10 Mbps x .7	350	\$8,000
T3	45 Mbps	2,700	\$30,000
100 Base-T LAN	100 Mbps x .7	4,000	N/A

Table 6-1 Número estimado de usuarios y el costo para soportarlos en una base 24/7 por cada tamaño de conexión de Internet, suponiendo un flujo de bits codificados de 16kbps (usando unicast)

Analizando la tabla puede observarse que el proveedor de servicio puede transmitir algunos miles de usuarios antes de llegar a su máxima capacidad de una conexión muy costosa. Agregando a esto el costo por mes de soporte por cada uno de estos usuarios con acceso a Internet desde el servidor. La operación sería bastante costosa cuando se considera que el potencial de servicio es para todo el mundo.

Accesibilidad y portabilidad. Para escuchar una transmisión de audio vía Internet se requiere de una computadora que tenga acceso a Internet.

6.3.1 Tecnología Webcasting

La tecnología de Web-audio se desarrolla en dos grandes áreas: La compresión de datos (audio digital) y el método de streaming. Aplicando la compresión de datos a él audio vía Internet permitirá a señales de audio de marginal a fidelidad razonable pasar a través de pequeños incrementos de anchos de banda. Una señal estéreo con calidad de CD requerirá de una tasa de transferencia de datos de alrededor de 1.5Mbps suficiente para llenar una conexión de Internet T1. Los sistemas de compresión de datos minimizan estos requerimientos por medio de la reducción de señales de audio digital a tasas de transferencia que pueden ser transportadas en típicas conexiones de Internet.

Streaming permite que los procesos packet-switched de Internet sean usados para que parezcan continuos, es decir reproducción de audio en tiempo real. La primera implementación a gran escala de streaming audio fue implementada en el sistema *MBone (Multimedia Backbone)* usando una conexión de Internet LAN de alta velocidad y estaciones de trabajo UNIX. La compresión de audio

usada en aplicaciones como Nevot y VAT fue una implementación primitiva de *PCM (Pulse Code Modulation)* prácticamente idéntico al usado en el sistema telefónico dial. Se tenía entonces la calidad auditiva del teléfono, que consumía alrededor de 78kbps de ancho de banda.

El primer sistema comercial del mercado fue el realizado por Progressive Networks llamado Real Audio 1.0. En este sistema la compresión de audio fue distribuida a usuarios conectados con un Modem a 14.4 kbps, pero la calidad fue pobre. La versión 2.0 requería un MODEM a 28.8 kbps, el cuál ofreció una mejor calidad pero no se aproximaba a la alta fidelidad.

Alrededor del año de 1996, la comunidad de audio profesional comenzó a adecuar su trabajo en la codificación de audio perceptible para Internet. La investigación en éste campo fue iniciada para aplicaciones de audio digital y transmisiones de televisión. Fue implementada en productos como el sistema de Satélite DirecTV, los reproductores Sony Minidisc y en sistemas de codificación profesionales ISDN para transmisión remota. Las tecnologías involucradas en estos productos incluían la familia MPEG, Dolby AC-3 y los algoritmos Lucent's PAC. Productos de Internet como Macromedia Shockwave audio y Microsoft Netshow usados para transmisión de streaming usan MPEG-2 Layer 3, mientras que RealAudio 3.0 usa Dolby AC-3.

Mientras los algoritmos de codificación perceptual tengan una mejoría significativa sobre las aproximaciones anteriores, las señales de audio con bajas tasa de transferencia requeridas para conexiones con Modem todavía carecen de altas frecuencias y son un poco ruidosas y/o distorsionadas. La baja fidelidad de audio en Internet no sorprende a profesionales del audio: La tasa de reducción de datos requerida para conexiones de Modem a 28.8 kbps es 50 a 1 o más, y no se logra sin una desventaja audible. MPEG Layer 3 fue originalmente pensada para mono bit rates de por lo menos 56kbs y en Dolby AC-3 para 128 kbps. Para las conexiones más rápidas las proporciones tasa-reducción son de 12 a 1 sin una degradación audible. La mejora del codec MPEG Layer 3 logra una calidad FM estereofónica a 56 kbps y una fidelidad indistinguible de CD a 128 kbps. De está manera al incrementar la velocidad de la conexión, el audio a través de Internet se aproxima a los niveles tradicionales de radiodifusión incluso en modems con tasas de transferencia de 28.8 kbps.

Las técnicas de procesamiento especial como la compresión y ecualización del rango dinámico multibanda usadas para mejorar las señales de audio cuando se transmiten al aire pueden ser aplicadas a las señales antes de la conversión digital y la compresión de datos con el fin de disminuir la capacidad audible de los artefactos de compresión.

6.3.2 La transmisión de Internet

El problema principal con las transmisiones de audio a través de Internet es que el sonido en tiempo real requiere un flujo continuo de bits, y los diseñadores de Internet solo se interesaron por el texto y los datos gráficos. Además los routers de Internet simplemente disminuyen paquetes cuando encuentran congestión.

El trabajo realizado para manejar audio y video streaming en este entorno se ha concentrado principalmente en protocolos de bajo nivel que soporten el transporte de paquetes. El protocolo de transmisión más común de Internet es TCP/IP, un protocolo de dos capas. El IP (Internet Protocol) describe el nivel más bajo, y es simplemente el método para formar y asignar ruta a los paquetes; El TCP (Transmisión Control Protocol) agrega tres funciones críticas:

- **La secuencia de paquetes.** El TCP numera cada paquete de modo que puedan ser reensamblados propiamente en el receptor.
- **Confiabilidad.** El TCP garantiza que todos los paquetes finalmente llegarán a tiempo al pedir la retransmisión de paquetes perdidos. Esto corrige el problema original, pero suma retardos.
- **Control de flujo.** Internet fue diseñado sobre el principio del buen vecino. Cuando se llega a congestionar, los clientes están a la espera afuera simultáneamente para que todos obtengan una porción justa de la capacidad disponible. Así el TCP es responsable en la reducción de flujo de bits. También proporciona una salida lenta en la cual la velocidad de comunicación de las rampas es lenta al comienzo de una nueva conexión de manera que los paquetes no son enviados tan rápido como el ancho de banda permitido del usuario.

Streaming media puede simplemente usar el TCP como un protocolo de transmisión. Los productos streaming usan este método de transmisión. Estos sistemas utilizan suficiente buffer en el lado del receptor para eliminar los retrasos potenciales del TCP, los cuáles no son un problema en transmisiones donde no se requiera interactividad.

Una importante ventaja de este método es en el flujo de datos en vivo (live stream) los cuales pasaran a través de cualquier barrera que permita un navegador normal web. Sin embargo el flujo de control del TCP podría afectar a los usuarios cuando compiten por el ancho de banda con usuarios que no lo usan, esto se reducirá cuando la capacidad de internet se incremente.

Un método alternativo es el realizado por Robust UDP, el cual está por debajo del Real Audio 3.0 de Progressive Networks. El User Datagram Protocol (UDP) es el nivel de acceso mas bajo posible a los paquetes IP, permitiendo a un desarrollador diseñar un uso confiable y una capa de control. PN de Robust UDP proporciona clasificación de paquetes y confiabilidad con ningún mecanismo de flujo de control. Progressive Networks también ha introducido *Real Media Architecture* (RMA), una plataforma abierta para desarrolladores de aplicaciones streaming media en tiempo real, así como un mercado y canales de distribución para productos y contenidos streaming.

6.3.3 IP Multicast

Para que Internet sea un medio viable de audio en tiempo real, necesita un método para servir a grandes audiencias. Acceder a una IP Multicast, una colección de herramientas para dirigir el costo de ancho de banda, la disponibilidad y los problemas de calidad-servicio que enfrenta el tiempo real y un Webcasting de gran escala.

En vez de duplicar los datos, el multicast manda la misma información sólo una vez a múltiples usuarios. Cuando un usuario pide un stream, los routers de Internet encuentran el nodo más cercano que tenga la señal y lo reproduce, haciendo un modelo escalable. Los multicast IP funcionan sobre cualquier tipo de red que acarree IP, incluyendo la red *Asynchronous Transfer Mode (ATM)*, la *frame relay*, la *dialup* e incluso enlaces satelitales. Originalmente desarrollada a finales de los años 80s, hoy en día es soportada por la mayoría de vendedores de *Internetworking* y está tomando velocidad.

La confiabilidad es un problema con el multicast porque no hay necesariamente un camino bidireccional desde el servidor al usuario para soportar la retransmisión de los paquetes perdidos. Aunque estén ahí, una serie de paquetes perdidos podrían crear suficiente tráfico de retorno para invalidar el ahorro de ancho de banda multicast. Por esta razón TCP/IP no puede usarse.

Entre los protocolos de transporte desarrollados para IP multicast se tiene el protocolo RTP (RealTime Transport Protocol) y el RTCP (Realtime Transport Control Protocol) son de los principales para entregar multimedia en tiempo real. RTP agrega a cada paquete cabecera la información distribuida necesaria para la secuencia de datos y la sincronización. No proporciona los mecanismos para asegurar la entrega oportuna ni proporciona garantías de calidad de servicio. Tampoco garantiza la entrega, ni asume que la red subyacente sea confiable.

Todas las aplicaciones de IP Multicast corren sobre la capa de transporte UDP, el cuál provee solo detección de error. RTP y RTCP no ofrecen ninguna

garantía de servicio: RTCP proporciona una forma para rastrear el desempeño, pero no puede imponer un nivel de desempeño.

RTCP es indeciso sobre el contenido de la capa de transporte, pero la mayoría de las implementaciones usarán probablemente RTP. Sin embargo, ninguno de estos aborda el problema de la confiabilidad permanente. ¿Qué se puede hacer?

Aceptar la pérdida de paquetes y hacer que el decodificador de audio enmascare el efecto de la pérdida de datos. Dejar que el decodificador enmascare la pérdida de datos sería una solución maravillosa si se pudiera lograr. Desafortunadamente, con la reducción de datos necesaria para comprimir audio a través de internet, cada paquete transporta mucha información y su pérdida es difícil de compensar. Un audio continuo requiere una capa de transporte confiable; no obstante técnicas básicas de encubrimiento como la repetición del dominio de la frecuencia combinado con paquetes intercalados trabajan razonablemente bien si la pérdida de paquetes es mínima y se pueden tolerar desviaciones incidentales de la perfección.

Transmitir datos redundantes para corrección de error posterior. Agregando algunos datos redundantes se mejora considerablemente su desempeño; combinado con el entrelazado, está puede ser una buena estrategia pero requiere más ancho de banda para un nivel de calidad dado. Esto basta en una conexión de Modem a 28.8 kbps, donde necesitas desesperadamente todos los bits que puedas conseguir.

Asignar y garantizar suficiente ancho de banda de la red para que ningún paquete se pierda. Para asegurar la entrega a través de la red. El protocolo *RSVP (Resource Reservation Protocol)* permite a los usuarios pedir una calidad específica de servicio para un flujo particular de datos. Los usuarios pueden especificar que tanto ancho de banda van a necesitar y que retardo máximo pueden soportar; después dispositivos de internetworking (conexión de redes) reservan el ancho de banda para ese flujo. Un usuario se le concede cualquier canal que haya solicitado o se le da una señal de ocupado.

Las redes y los host RSVP trabajan en conjunto para lograr una garantía de calidad de la transmisión end to end. Todos los hosts, routers y otros elementos de infraestructura de red entre el receptor y el remitente deben soportar RSVP. Cada uno de ellos reserva recursos del sistema como ancho de banda, CPU y almacenamiento temporal de memoria para satisfacer la demanda.

RSVP opera sobre IP (IPv4 o IPv6), ocupando el lugar de un protocolo de transporte en la pila protocolar, pero proporciona servicios de session-layer (no transportan ningún dato). El protocolo RSVP es usado por los routers para

entregar solicitudes de control a todos los nodos a lo largo de las rutas de los flujos.

Los vendedores tienen implementado ambos RSVP por encima y por debajo de Winsock. Las aplicaciones RSVP-aware (conciente) pueden desarrollarse con Winsock 2, el cual tiene un API QoS-sensitive (sensible). Otro acercamiento es usar un proxy RSVP que funcione independientemente de la aplicación real, haciendo reservaciones RSVP.

RSVP plantea preguntas significativas sobre el envío para el ancho de banda de Internet. En el modelo actual, los ISPs venden con exceso su capacidad disponible y los consumidores aceptan retrasos. Desde que la reserva de recursos coloca una demanda específica sobre el ancho de banda, la sobreventa daría como resultado inaceptables señales ocupadas. Los ISPs ofrecerán probablemente diferentes niveles de servicio, y los servicios Premium serán cargados para reservaciones RSVP. El envío a través de múltiples carries también tendría que resolverse, así como la asignación de recursos computacionales a los reuters para inspeccionar y manejar paquetes sobre una base priorizada. Es incierto si los routers existentes podrían manejar una implementación a gran escala de RSVP en todo Internet.

Implementar un multicast especial capaz de retransmitir confiablemente cuando este disponible un camino bidireccional. Esto tal vez puede ser posible cuando haya un camino de retorno al servidor. En el más simple acercamiento, un receptor preguntaría solamente por una copia de cualquier paquete perdido y lo recibiría en un canal lateral (alternativo). Sin embargo, si la pérdida afecta a un gran número de receptores, la petición de reconocimiento negativo (NAK) podría hundir al servidor y tomar tanto ancho de banda como en las múltiples conexiones individuales unicast. Varias ideas se han propuesto alrededor de este inconveniente, pero es incierto si alguien a fin de cuentas podría escalar a proporciones de Internet mundiales.

6.3.4 Protocolo Streaming de Tiempo Real

El crecimiento de multimedia en tiempo real en la red ha agotado las capacidades limitadas de HTTP en esta área, el grupo especializado de Ingeniería en Internet (IETF) está intentando estandarizar funciones como el comienzo y la detención de streams, la sincronización de múltiples elementos de medios e implementar otros controles. El trabajo principal se incluye en el Real-Time Streaming el Protocolo (RTSP), el cual fue propuesto por Progressive Networks, Netscape, y la Universidad de Columbia en octubre de 1996.

La última versión de RTSP se apropia fuertemente de HTTP y esencialmente provee servicios de nivel HTTP para tipos de datos streaming en tiempo real. Sin embargo, difiere fundamentalmente de HTTP en esa entrega de datos que toma lugar fuera de la banda, en un protocolo diferente. El RTSP establece y controla cada uno o varios streams sincronizados en tiempo de medios continuos. Se espera que RTSP use TCP como capa de transporte (para control solamente), pero UDP tal vez sea soportado opcionalmente. Aunque esta especificación de proyecto está en fases tempranas de sumisión en el IETF y seguramente sufrirá cambios significativos, las implementaciones están disponibles actualmente.

RTSP puede usar mecanismos de seguridad Web, en la capa de transporte o dentro del propio protocolo. Todos los mecanismos de autenticación HTTP como la básica y la asimilada son aplicadas directamente. Los problemas de licencias de contenido y la administración de derechos no son directamente tratados en el proyecto actual, pero quienes lo proponen dicen estar trabajando en mecanismos para cumplir esta tarea.

6.3.5 El método Microsoft

Visiblemente ausente de la lista de partidarios de RTSP es Microsoft, quienes han implementado un protocolo rival llamado Active Streaming Format (ASF) en su plataforma servidor Netshow. Ofreciendo una capacidad similar a RTSP, la documentación de Microsoft se refiere modestamente a ASF como un formato de archivo, y lo describe como un componente de la estrategia global Active X de Microsoft. En realidad, es un tipo de archivo o carpeta que empaqueta múltiples "media objects" en un armazón unificado. Como RTSP, puede usarse para sincronizar un número de objetos multimedia como: audio, video, imágenes, eventos, URLs, páginas HTML, comandos script y programas ejecutables. A Diferencia de RTSP, un stream ASF incluye control y elementos de contenido.

Una función inusual de ASF es la opción de emplear una variedad de corrección de errores y técnicas de encubrimiento ó ocultación. El método redundante de corrección de errores posterior combinado con el entrelazado permite recuperar los datos perdidos sin la retransmisión cuando la pérdida de los datos no es demasiado grave. El cambio es el requisito de información adicional inherente en cualquier sistema que agrega información redundante.

Ya que ASF simplemente describe un formato de archivo, puede transportarse en una variedad de capas diferentes de red de nivel bajo, incluyendo TCP/IP, UDP/IP, y RTP. De hecho, Netshow soporta un modo de HTTP para usuarios que están detrás de firewalls que bloquean paquetes UDP.

Conclusiones

El desarrollo de métodos de compresión, así como el de las redes ha hecho posible la transmisión de audio y video digital a través de Internet. Esto ha permitido que se manipule y controle la información desde una computadora. Tal desarrollo no ha terminado, continuamente se siguen perfeccionando los algoritmos de compresión y las redes de comunicación son cada vez mejores para lidiar con el tráfico multimedia. Pero uno de los principales problemas que tiene el audio y video en Internet son las redes que lo soportan, las cuales no fueron diseñadas para tratar con tráfico multimedia. Además los protocolos de red empleados no garantizan un flujo continuo de bits, que son requeridos para aplicaciones multimedia en tiempo real.

Aunado a esto el ancho de banda empleado por el usuario final limita la reproducción y calidad de una transmisión de audio o de video. Es cierto que existen servicios de Internet que proporcionan mayores anchos de banda, lo cual permite tener una mejor calidad pero a un mayor costo y no soluciona por ejemplo el problema del tráfico. Los métodos de compresión en este caso se adecuan al canal de información del usuario, es decir, presentan la calidad conforme al ancho de banda disponible.

En Internet se utiliza generalmente la compresión que se conoce como lossy (con pérdida), ésta permite reducir una gran cantidad de información redundante de los formatos nativos de audio y video. Una de sus ventajas es que sus pérdidas son imperceptibles para el oído o la vista del usuario, logrando facilitar la distribución de grandes cantidades de información por Internet aún en conexiones pequeñas. Muchos de los estándares utilizados en aplicaciones de telefonía, videoconferencia, audio, video, imágenes y televisión digital son aplicados a las redes. De ahí que, la compresión haya fomentado tanto aplicaciones multimedia como de consumo.

El uso de nuevas tecnologías como Webcasting, que involucra servidores dedicados a la transmisión de audio y video en tiempo real acarrea un mayor

CONCLUSIONES

costo a los proveedores de este servicio, ya que cada usuario se conecta a un servidor individualmente. Esto significa que el proveedor de servicio tiene que pagar la capacidad del servidor para transmitir a usuarios simultáneos y a su vez la conectividad a todos los usuarios en conexiones separadas. En este sentido la rentabilidad y el costo limitan un potencial de servicio global.

Ciertamente existen inconvenientes técnicos y económicos que han retrasado el uso de estos servicios de forma global, pero se están desarrollando nuevas soluciones cada día, y la demanda comercial por los servicios resultantes es amplia y crece continuamente. Uno puede esperarse con base en los temas desarrollados en esta tesis una continua innovación tecnológica y un patrón de nuevos productos y servicios que faciliten las comunicaciones multimedia a través de Internet.

Bibliografía

Computer Communications Network

by N. Abramson, F. Kuo

Prentice Hall Trade; ISBN: 0131654314; (June 1973)

Multimedia Communications: Protocols and Applications

by Franklin F. Kuo (Editor), Wolfgang Effelsberg, J. j Garcia-Luna-Aceves,
Frank Kuo, J. J. Garcia Luna (Contributor)

Prentice Hall PTR; ISBN: 0138569231; 1st edition (September 26, 1997)

Multimedia Communications: Directions and Innovations

Jerry D. Gibson (Editor)

Academic Press, Incorporated; ISBN: 0122821602; 1st edition (October 2000)

Digital Compression for Multimedia: Principles and Standards

Jerry D. Gibson, Baker, Lookabaugh, Lindbergh, Tom Lookabaugh, Dave
Lindbergh, Berger, Richard L. Baker, Toby Berger

Morgan Kaufmann Publishers; ISBN: 1558603697; 1st edition (October 1997)

Glosario

A/D : Acrónimo de Analog to Digital (De analógico a digital). Conversión de un sistema a otro.

Whiteboard : Área de una pantalla de visualización en la que múltiples usuarios pueden escribir o dibujar.

Buffering : Área de almacenamiento temporal.

Burst : Conjunto de bits, bytes o caracteres agrupados juntos para transmisiones.

Codec : Abreviatura de compresor/decompresor, un codec es cualquier tecnología para comprimir y descomprimir datos. Los codecs pueden implementarse por software, hardware, o una combinación de ambos. En telecomunicaciones, (abreviatura de codificador/decodificador) un dispositivo que codifica o decodifica una señal.

Connectionless : Se refiere a los protocolos de red en los cuales un servidor puede mandar un mensaje sin establecer una conexión con el receptor. Es decir el host pone el mensaje en la red con la dirección destino y espera que este llegue.

Frame Rate : Frecuencia en que se representan fotografías en una secuencia de película.

Host : (Anfitrión). Ordenador, en la mayoría de los casos, con capacidad multiusuario y grandes recursos, al que se accede remotamente. Suele albergar gran cantidad de información o servicios telemáticos específicos accesibles en línea.

ISDN : Acrónimo de Integrated Services Digital Network (Red digital de servicios integrados).

Overhead : Uso de los recursos de la computadora para realizar una tarea específica.

Peer to peer : La tecnología Peer to Peer (P2P, de igual a igual) puede definirse como el uso coordinado de recursos geográficamente distribuidos en ausencia de control central, basado en intercambios directos de información.

Protocolo : Directrices que regulan las comunicaciones entre ordenadores.

Streaming : Técnica para transferir datos de tal forma que puedan ser procesados como un flujo estable y continuo.

Sink : Dispositivo que recibe, monitorea información u otras señales desde una fuente.

Throughput : Es la cantidad de datos transferidos de un lugar a otro o procesados en una cantidad específica de tiempo.

Time - Slot : Período de tiempo durante el cuál ciertas actividades son controladas por una reglamentación específica.