

Video Streaming

- Introducción -

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Programa de Tecnologías para la gestión distribuida
de la información

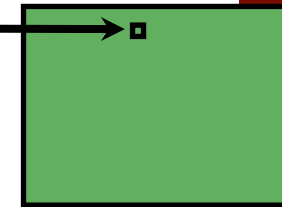
Contenido

- Características del vídeo
- ¿Streaming?
- Arquitectura del servicio
 - El servidor
 - La red

Vídeo

Vídeo digital

- Dibujar una imagen detrás de otra da sensación de movimiento
- Del orden de al menos 25 imágenes por segundo
- Cada fotograma (frame) es una imagen digital a transmitir
- Su calidad depende de
 - Resolución (pixels)
 - Colores (cuántos bits para representar el color, RGB o luminancia/crominancia)
- Ejemplo:
 - Vídeo analógico en PAL
 - 625 líneas (576 visibles) aspecto 4:3 25 fps
 - Usamos 768x576 pixels, en cada pixel representamos 16 millones de colores con 8:8:8 bits para RGB
 - Serían 1300 KB cada frame o sea **265Mbps** para enviar **un solo canal de TV**
- Es muy importante usar técnicas de compresión en los formatos de vídeo digital



Características

Audio

- CD: 1.411 Mbps
- MPEG-1 Part 3 Layer 3 (MP3): 96, 128, 160 kbps
- Internet telephony: 5.3-13 kbps

Vídeo (+audio):

- MPEG 1
 - Entornos libres de errores (CD-ROM, VCD)
 - 1.2 Mbps el vídeo, 256 Kbps el audio
 - Busca la mejor calidad dado un bitrate
 - Permite acceso aleatorio a un frame
 - Basado en H.261, imagen en macrobloques y DCT de los bloques
- MPEG 2
 - Broadcast TV, DVD
 - 2-15 Mbps (vídeo+audio)
 - Basado en DCT pero con compresion basada en correlacion temporal
 - MPEG-2 part 7 advanced audio codec AAC
 - Define encapsulado (transport stream TS) para enviar el vídeo sobre la red

ITU-T H.26x

- ITU-T H.261 “Video codec for audiovisual services at px64 kbits”
 - CIF (352x288), QCIF (176x144) (resoluciones de luminancia)
 - El resultado es un bit stream
 - Videoconferencia sobre ISDN
- ITU-T H.263 “Video coding for low bit rate communication”
 - Basado en H.261
 - sub-QCIF (128x96) , QCIF, CIF, 4CIF (704x576) and 16CIF (1408x1152)
 - Videoconferencia sobre POTS
 - Basado en H.261, MPEG-1 y MPEG-2

ITU-T H.262

- “Information technology - Generic coding moving pictures and associated audio information: Video”
- MPEG-2 Part 2
- Pensando en vídeo sobre ATM y HDTV
- Soporta *scalable video encoding* (ej: codificación espacial escalable mediante varias capas que van refinando la imagen)
- Tipos de frames:
 - Intra Coded Pictures (I-Pictures): sin referencia a otras imágenes
 - Predictive Coded Pictures (P-Pictures): emplea *motion compensated prediction* con la anterior I- o P- Picture
 - Bidirectional-predictive Coded Pictures (B-Pictures): relativas a anterior y posterior I- o P- Picture
 - La organización en secuencia es flexible en el estándar

H.264

- “Advanced video coding for generic audiovisual services”
- También MPEG-4 Part 10
- Puede dar la misma calidad que el codec de vídeo de MPEG-2 con la mitad o un tercio del bitrate

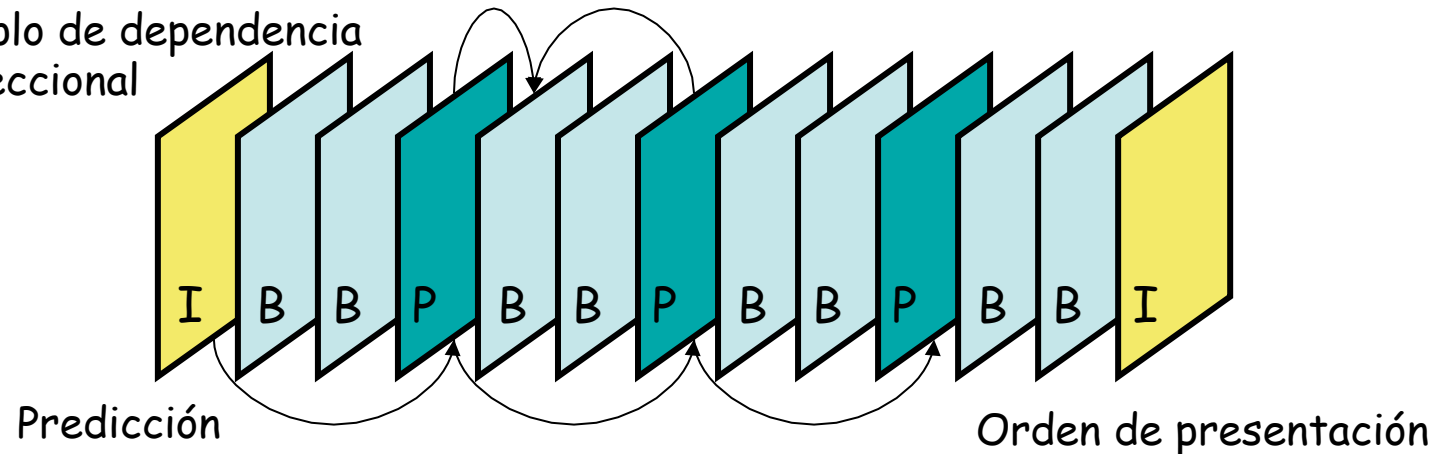
Use Scenario	Resolution & Frame Rate	Example Data Rates
Mobile Content	176x144, 10–15 fps	50–60 Kbps
Internet/Standard Definition	640x480, 24 fps	1–2 Mbps
High Definition	1280x720, 24p	5–6 Mbps
Full High Definition	1920x1080, 24p	7–8 Mbps

GoP: estructura

- *Group of Pictures*
- Típicamente 1/2 sec cada GoP
- Orden
 - De presentación
Ej.: IBBPBBPBBPBB ibbpbb...
 - De codificación
Ej.: I bb PBBPBBPBB i BB pbb...
- Closed or Open GoP
- Broken GoP: falta el GoP anterior

	I frame	P frame	B frame
Compression Ratio	Low	Good	Best
Random Access	Best	Hard	Hardest
Complexity	Normal	High	Highest

Ejemplo de dependencia
bidireccional

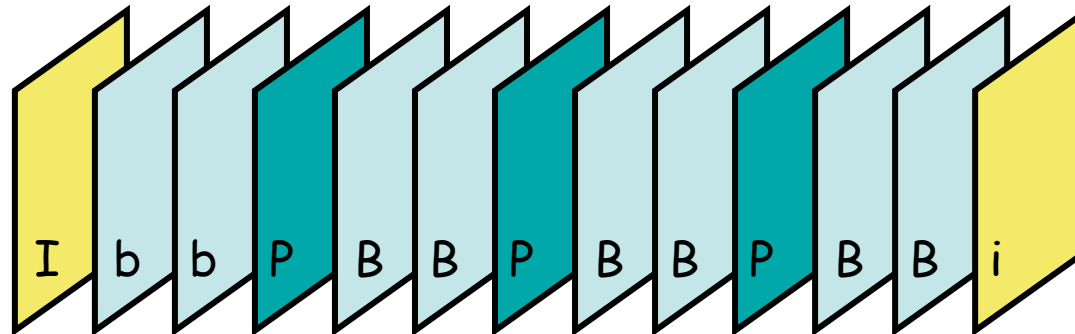


GoP: estructura

- *Group of Pictures*
- Típicamente 1/2 sec cada GoP
- Orden
 - De presentación
Ej.: IBBPBBPBBPBB ibbpbb...
 - De codificación
Ej.: I bb PBBPBBPBB i BB pbb...
- Closed or Open GoP
- Broken GoP: falta el GoP anterior

	I frame	P frame	B frame
Compression Ratio	Low	Good	Best
Random Access	Best	Hard	Hardest
Complexity	Normal	High	Highest

Dependencia unidireccional

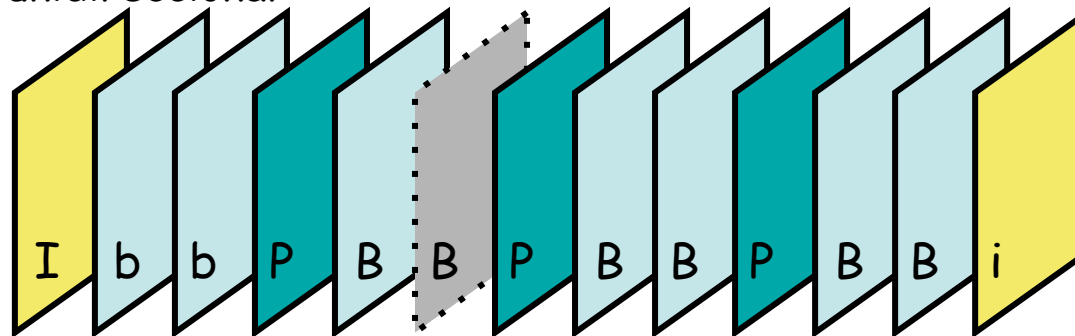


Orden de codificación/transmisión

GoP: pérdidas

- Pérdida de un B-frame
 - Sin repercusiones en otros frames

Dependencia unidireccional

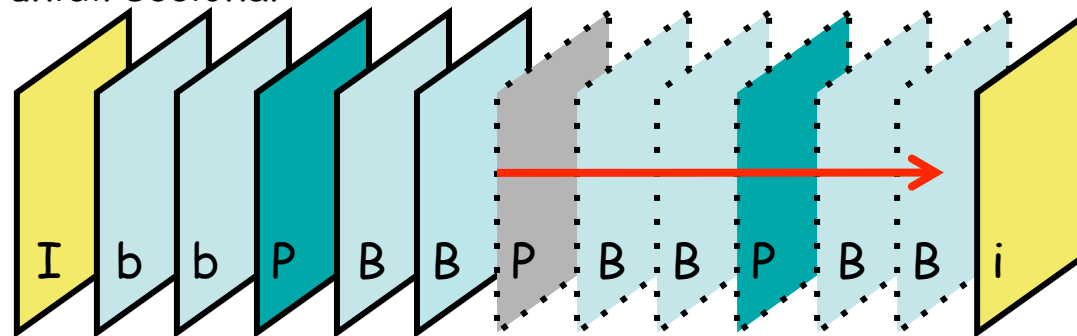


Orden de codificación/transmisión

GoP: pérdidas

- Pérdida de un B-frame
 - Sin repercusiones en otros frames
- Pérdida de un P-frame
 - No se podrán decodificar ni P- ni B- frames en el mismo GoP

Dependencia unidireccional

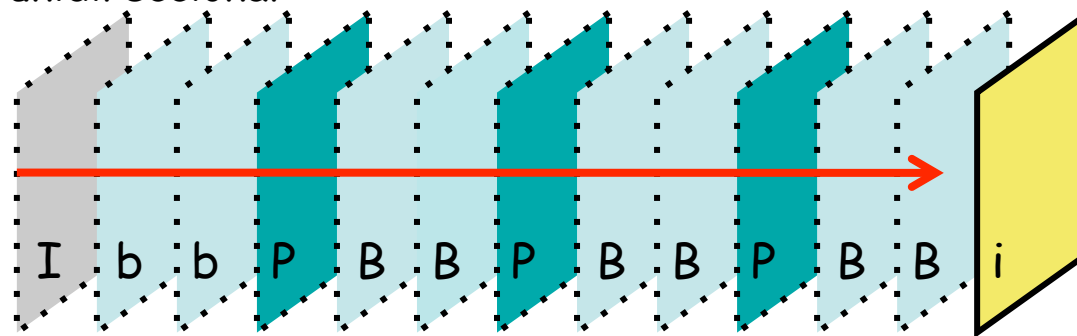


Orden de codificación/transmisión

GoP: pérdidas

- Pérdida de un B-frame
 - Sin repercusiones en otros frames
- Pérdida de un P-frame
 - No se podrán decodificar ni P- ni B- frames en el mismo GoP
- Pérdida de un I-frame
 - Tantos frames como el tamaño del GoP no se puede decodificar

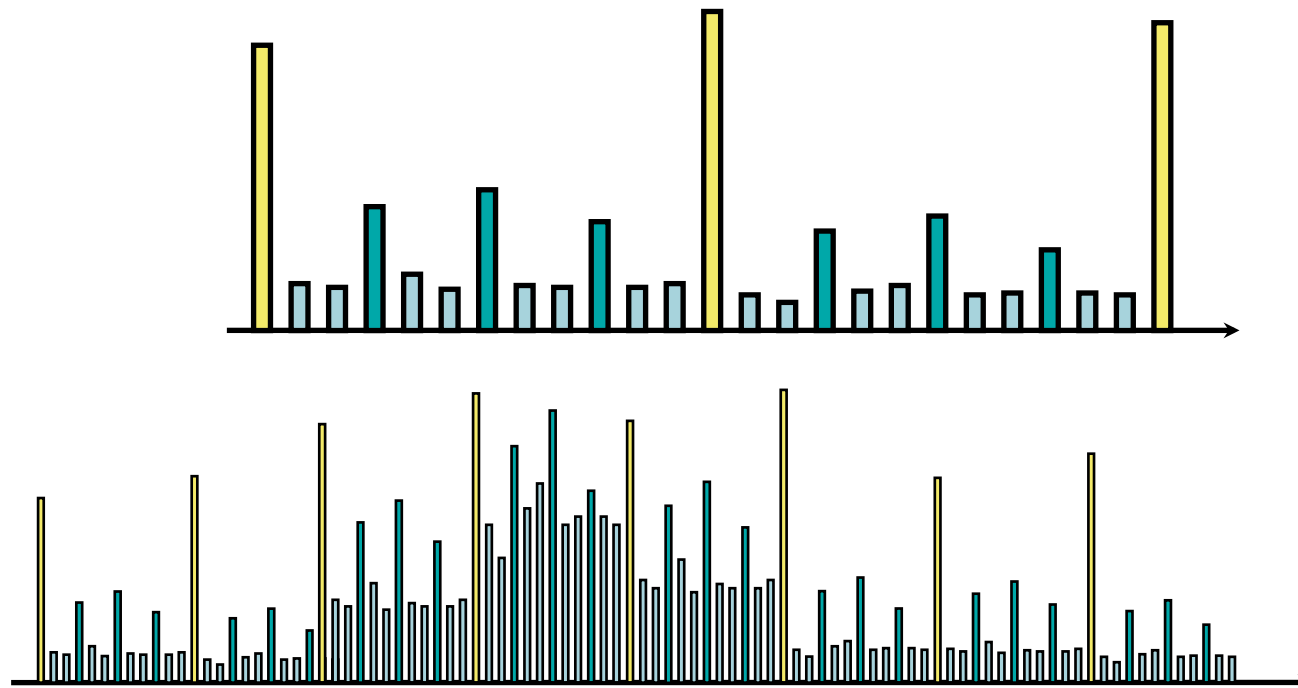
Dependencia unidireccional



Orden de codificación/transmisión

GoP: tráfico

- La cantidad de información a transmitir por cada frame es variable
- Las tramas I contienen más información
- Las tramas P y B son más pequeñas (mayor compresión)
- Variabilidad de tamaño en escalas pequeñas por las tramas
- Variabilidad de tamaño en escalas mayores según el movimiento de la escena



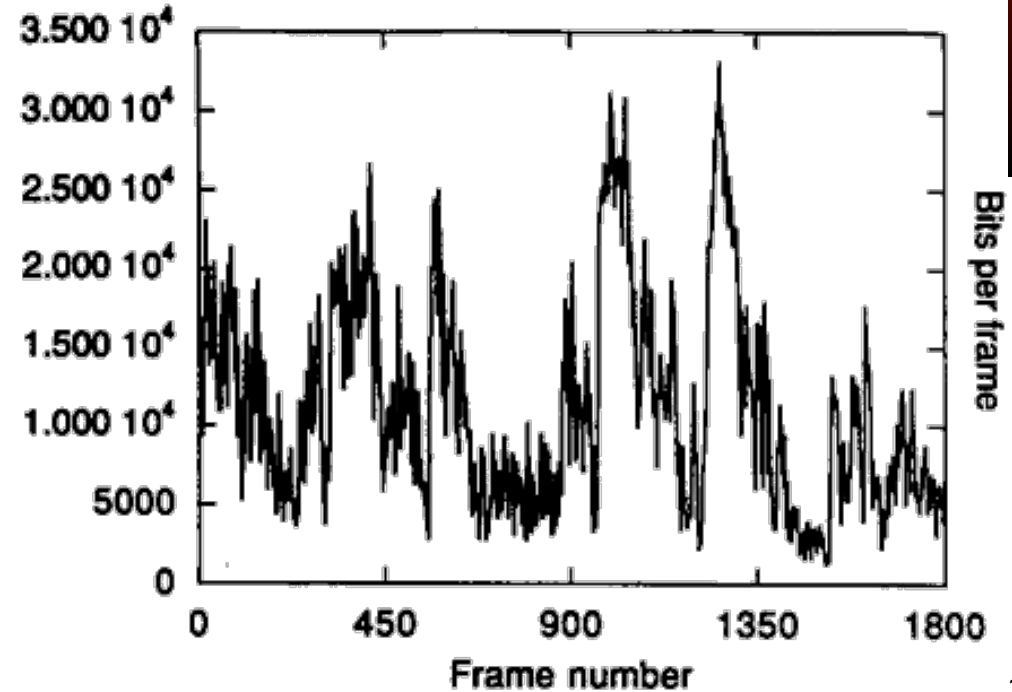
Tráfico generado por fuentes de vídeo

Interesa modelar:

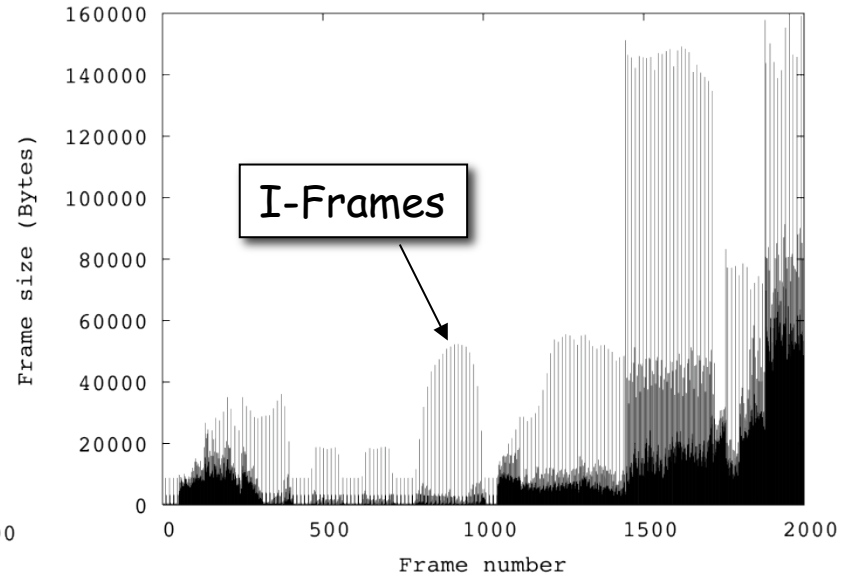
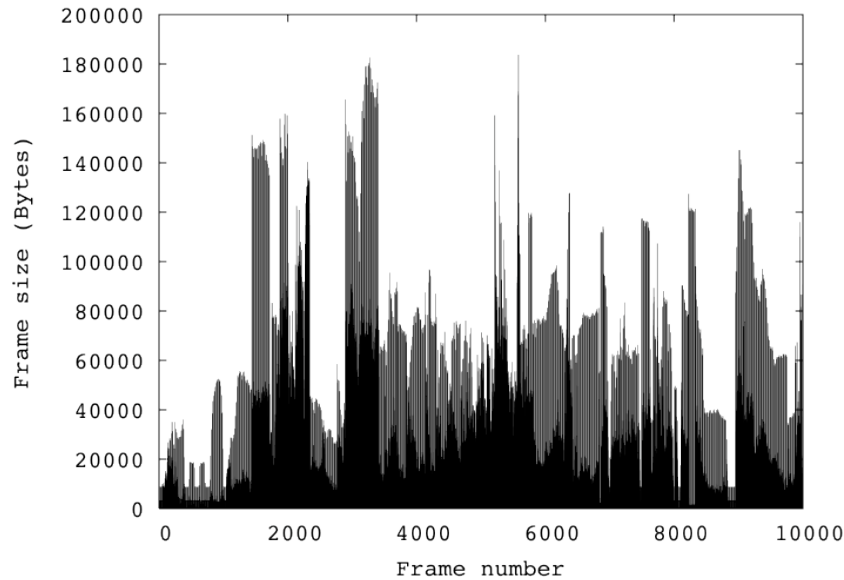
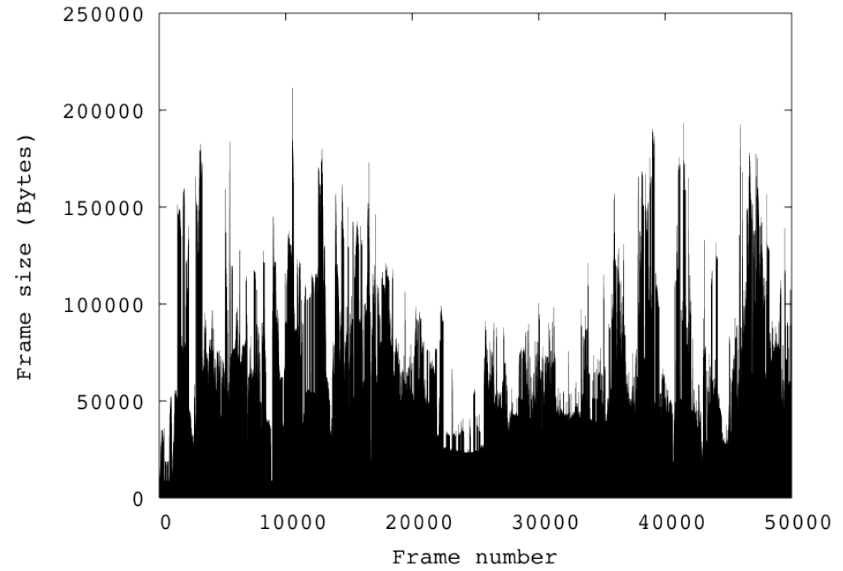
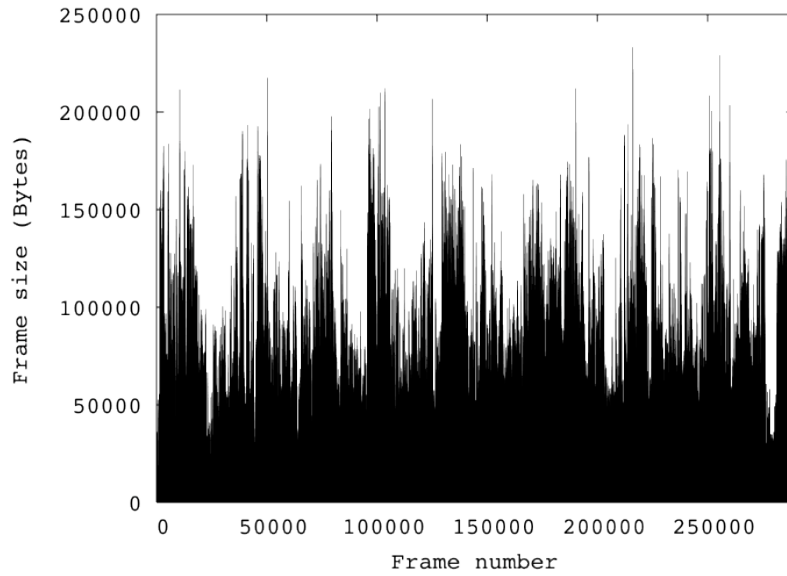
- PDF de tamaños de frames
- Autocorrelación

Para:

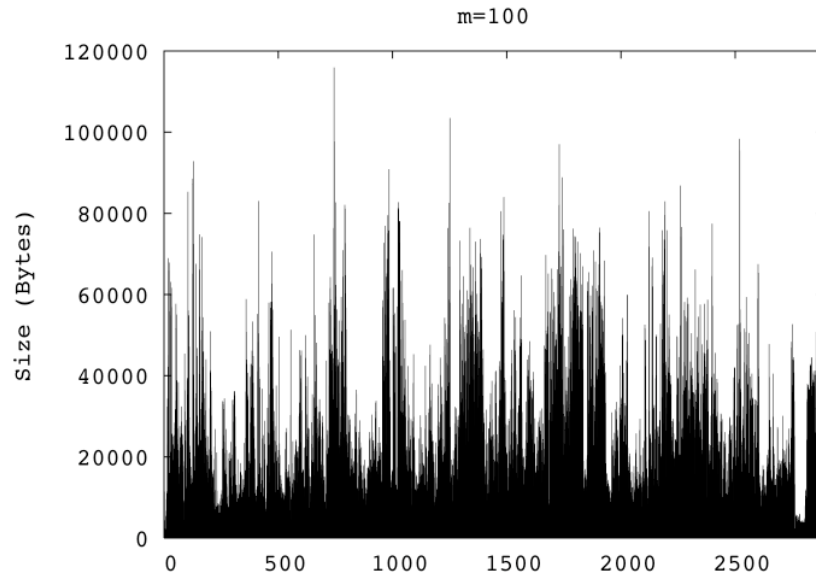
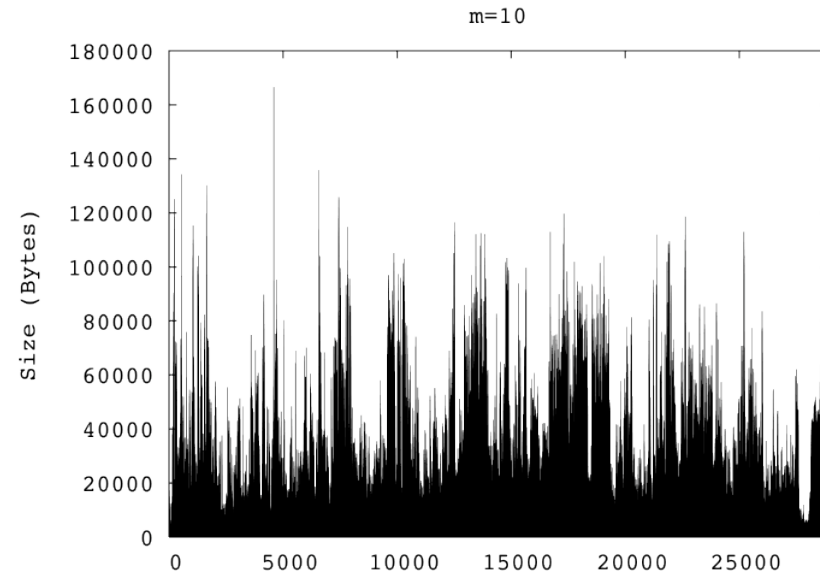
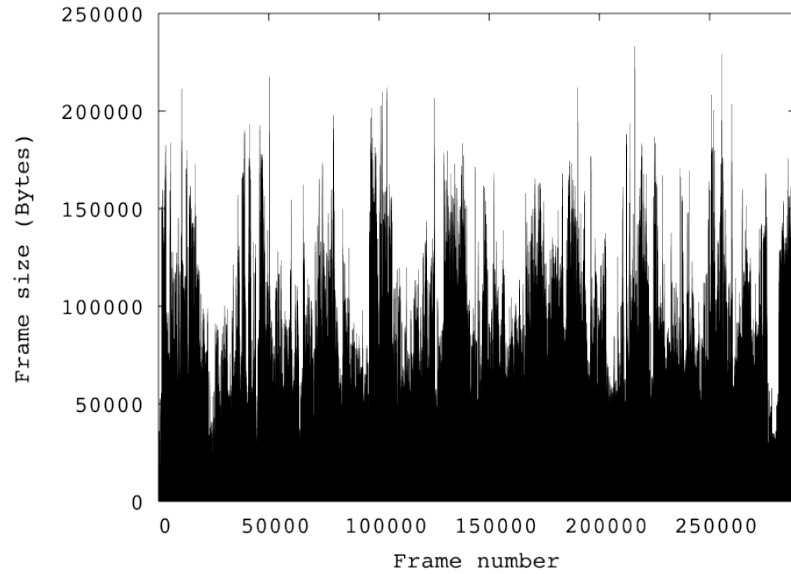
- Predecir pérdidas en buffers
- Dimensionar BW y buffer necesario



Ráfagas



En diferentes escalas



Multimedia Networking Applications

Características principales:

- Sensibles al retardo
 - end-to-end delay
 - delay jitter
- Toleran pérdidas: si son infrecuentes solo causan pequeños *glitches*
- Requisitos diferentes de la transferencia de datos (ficheros)

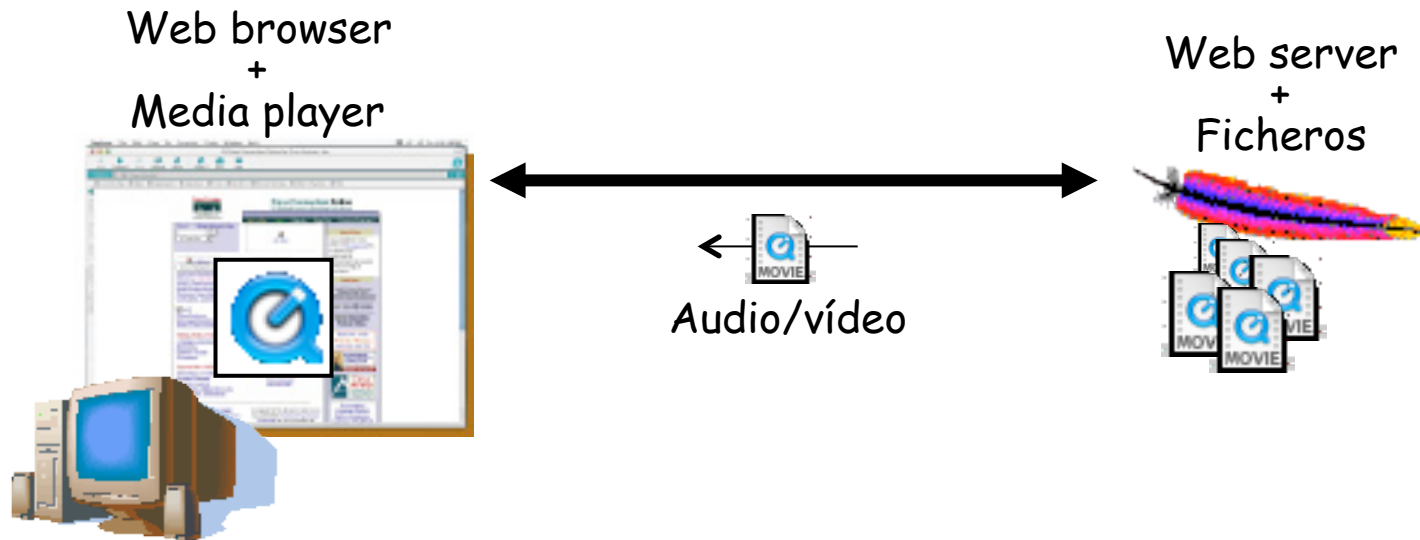
Clase de aplicaciones multimedia

- “Streaming” de audio o vídeo grabado
- Streaming de audio o vídeo en vivo
- Audio o vídeo interactivo

Jitter es la variabilidad de los retardos dentro del mismo flujo de paquetes

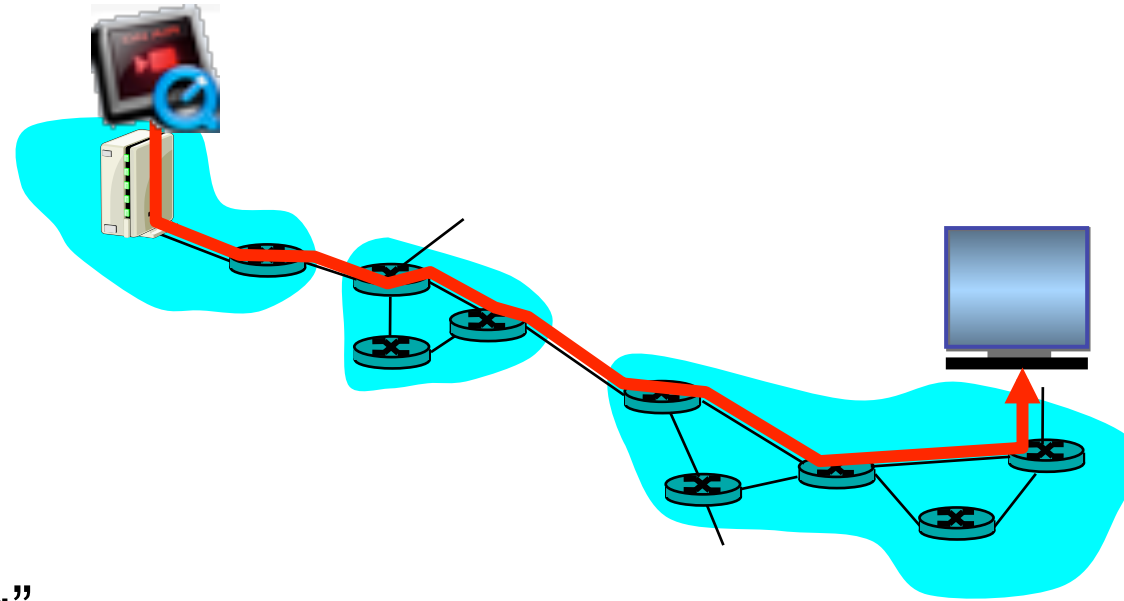
¿A qué llamamos *Streaming*?

Download+play vs “Streaming”



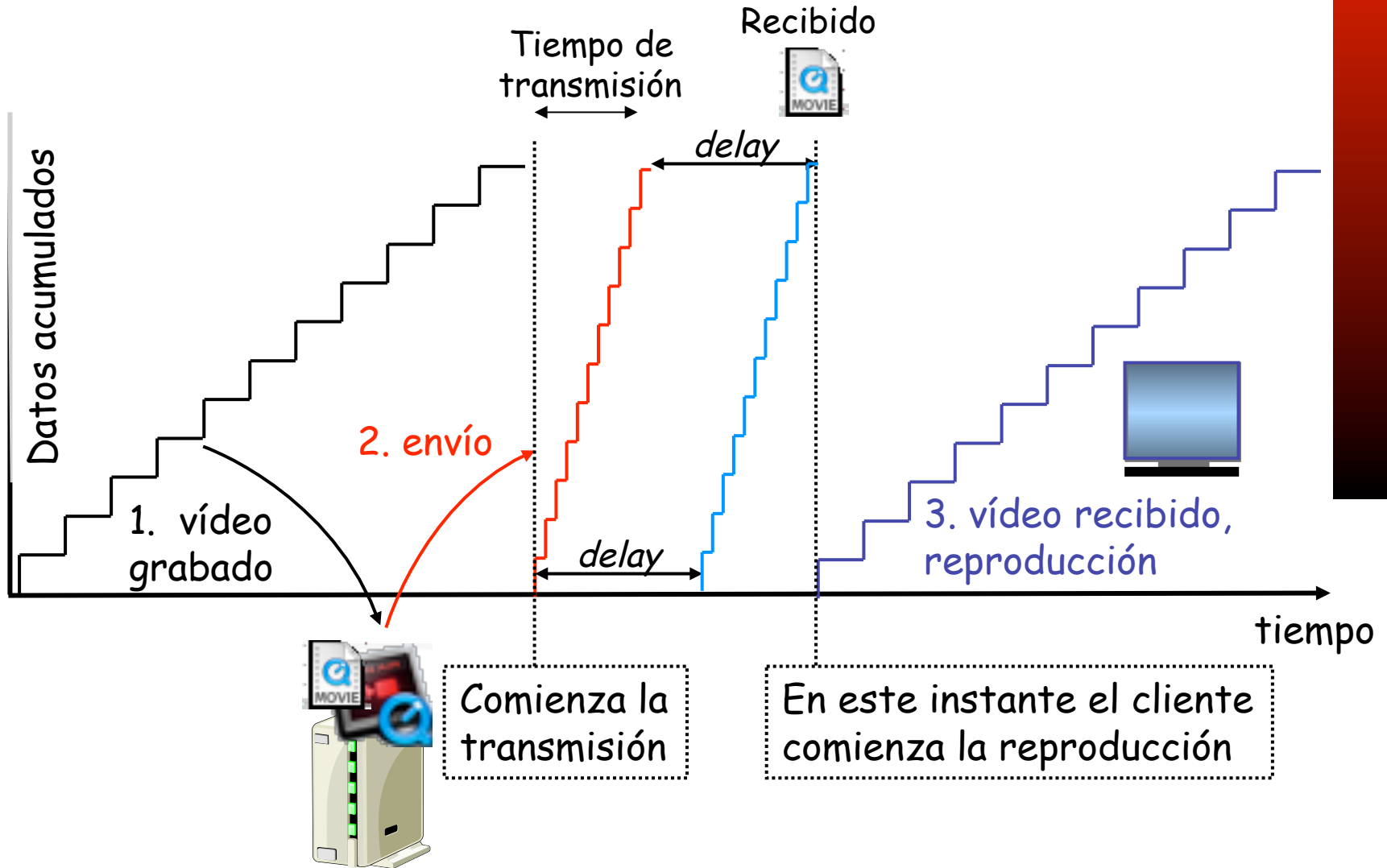
- Ficheros de audio o vídeo almacenados en servidor
- Ficheros transferidos como objetos HTTP
 1. Download+play
 - Recibidos completamente por el cliente
 - Pasados al reproductor
 - No hay “streaming”. Gran espera hasta empezar
 2. “Streaming”
 - Pasados al reproductor a medida que los recibe
 - Calcula cuándo empezar dada la velocidad a la que recibe
 - “Streaming” pero recibe a la máxima velocidad (transferencia HTTP)

“Streaming” de multimedia en disco

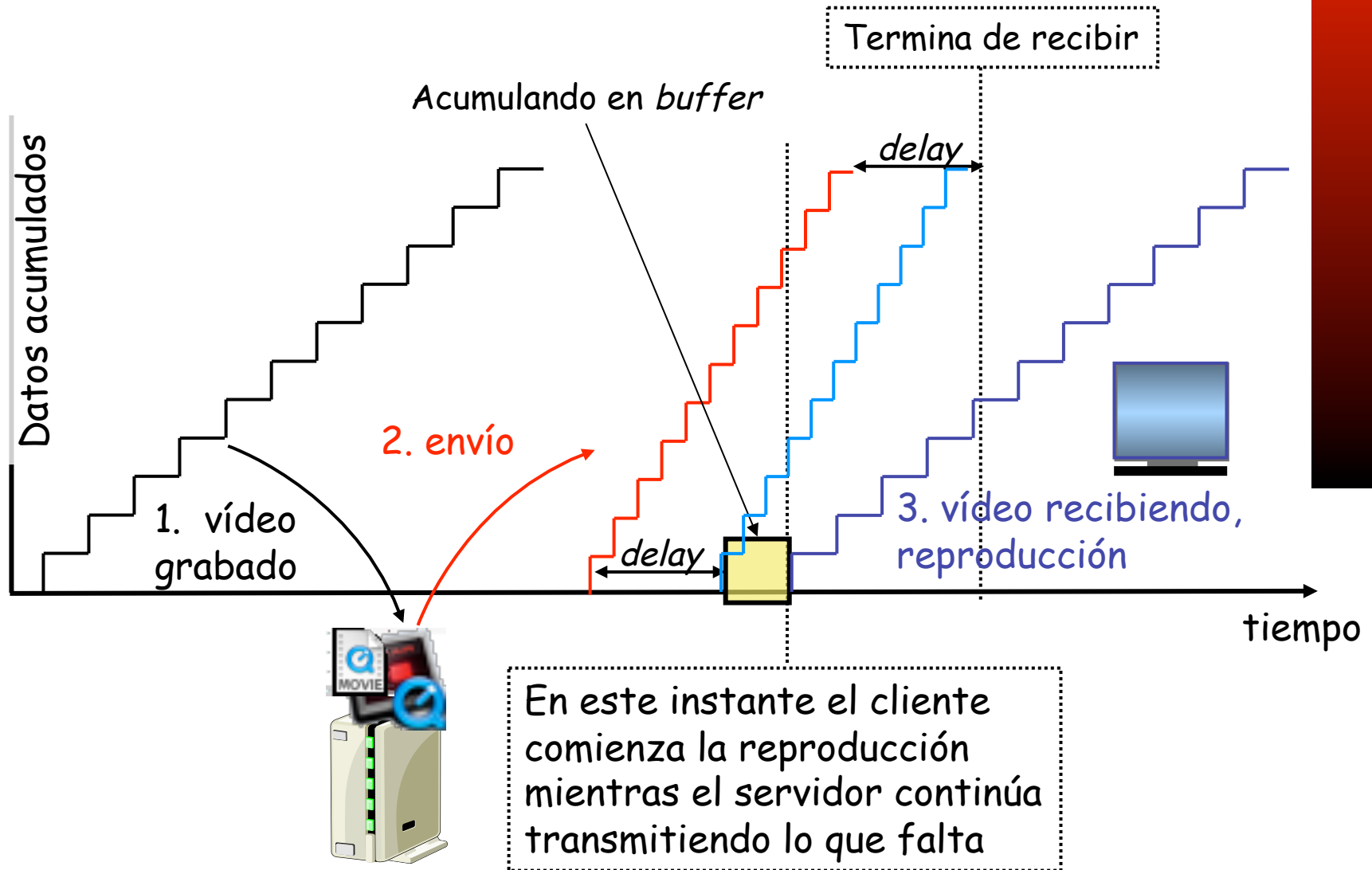


- “Streaming”
 - El cliente comienza la reproducción antes de recibir todos los datos
 - Es más bien un “play-as-you-get”
 - Requisitos temporales para los datos aún por recibir
- Streaming real
 - La velocidad de transmisión es aproximadamente la de consumir los datos

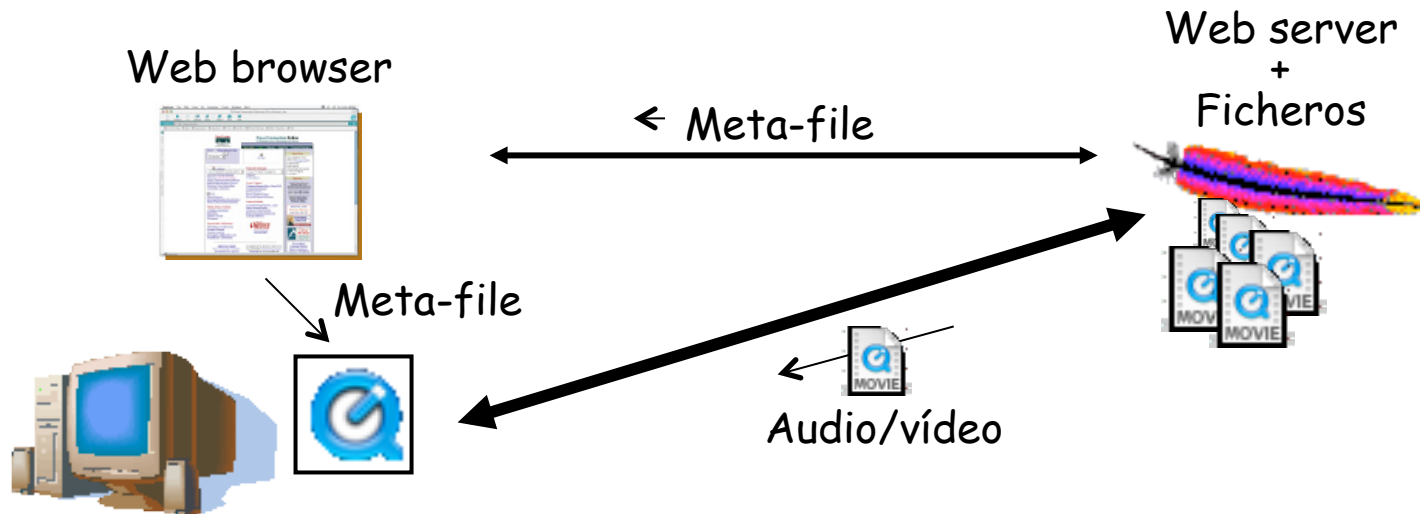
Download+play



Play-as-you-get

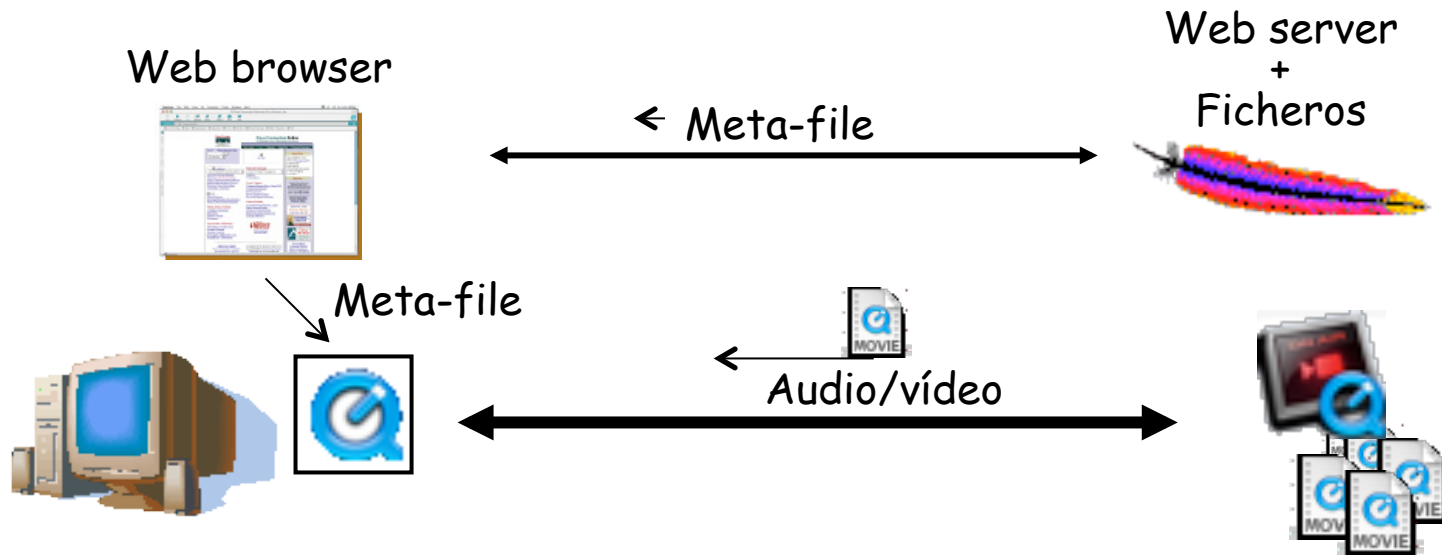


Play-as-you-get



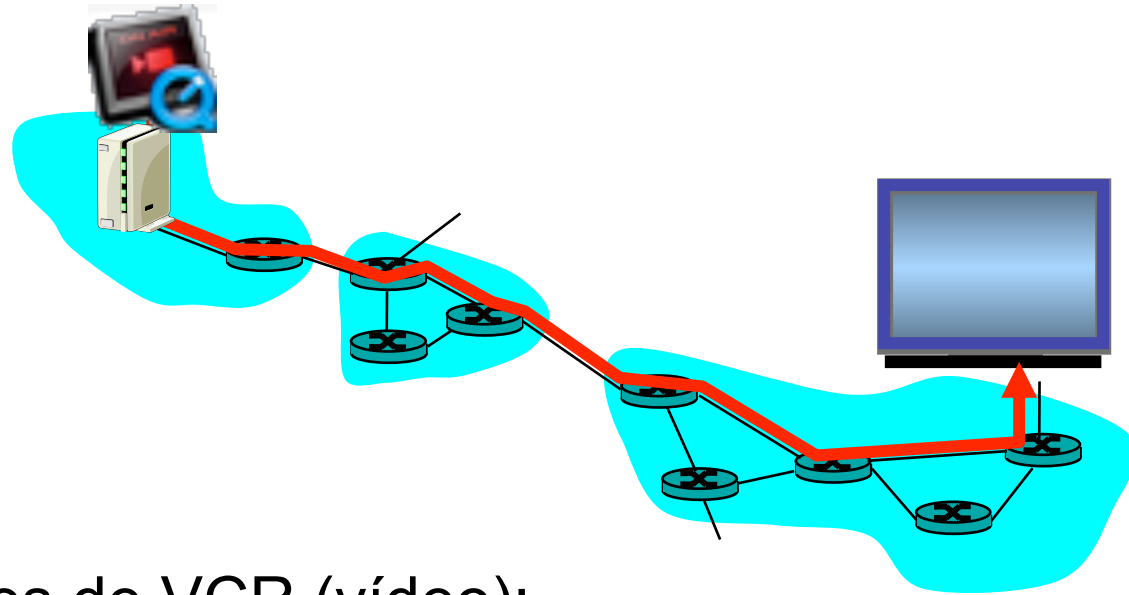
- Ficheros de audio o vídeo
- El navegador obtiene un *meta-file* que describe cómo conseguir la película
- Le pasa ese *meta-file* al reproductor
- El reproductor contacta con el servidor y solicita el fichero
- El servidor se lo envía (HTTP)

Play-as-you-get



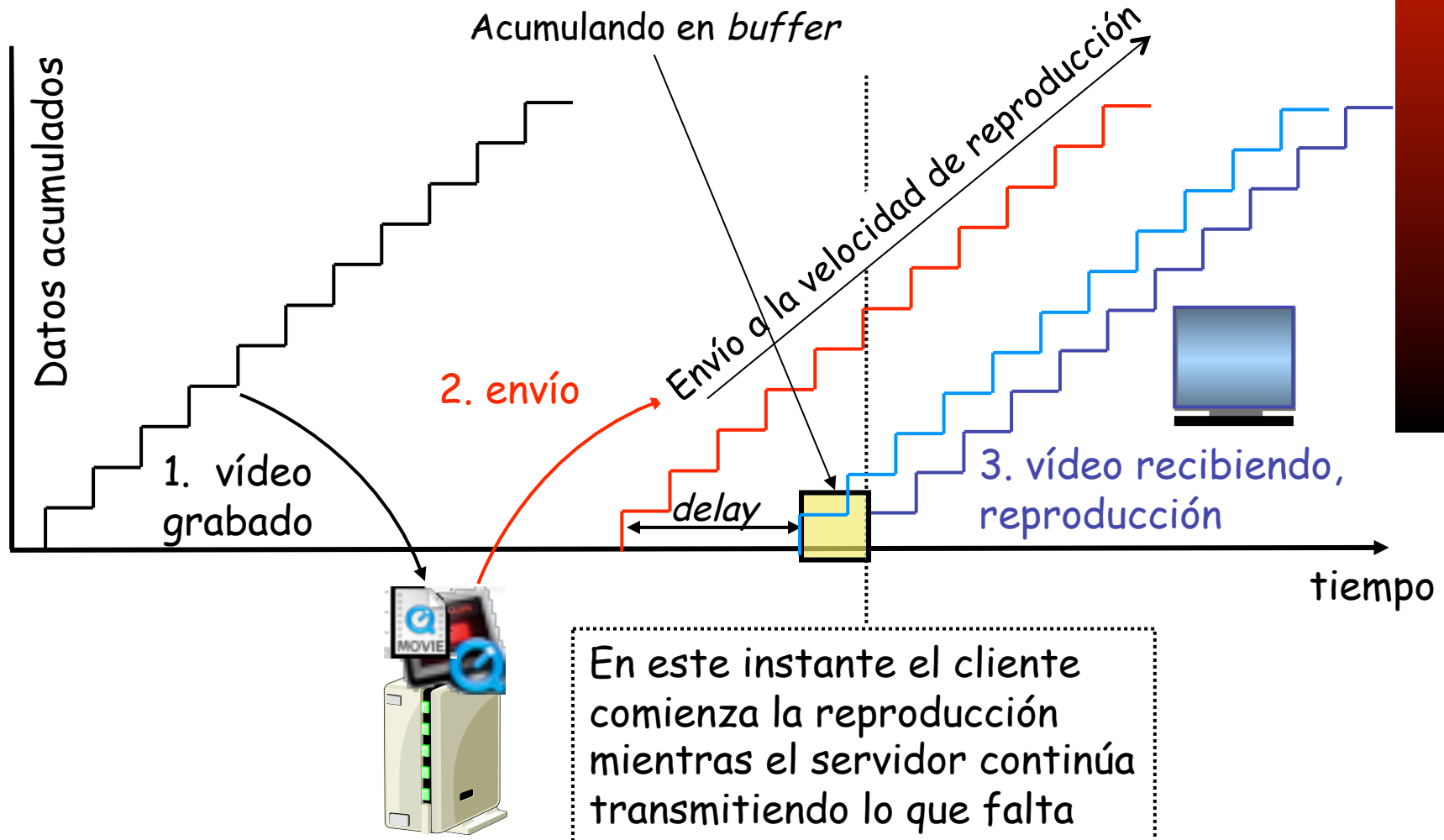
- Permite que se emplee un protocolo diferente de HTTP
- Permite emplear UDP en vez de TCP

“Streaming” de multimedia en disco: Interactividad

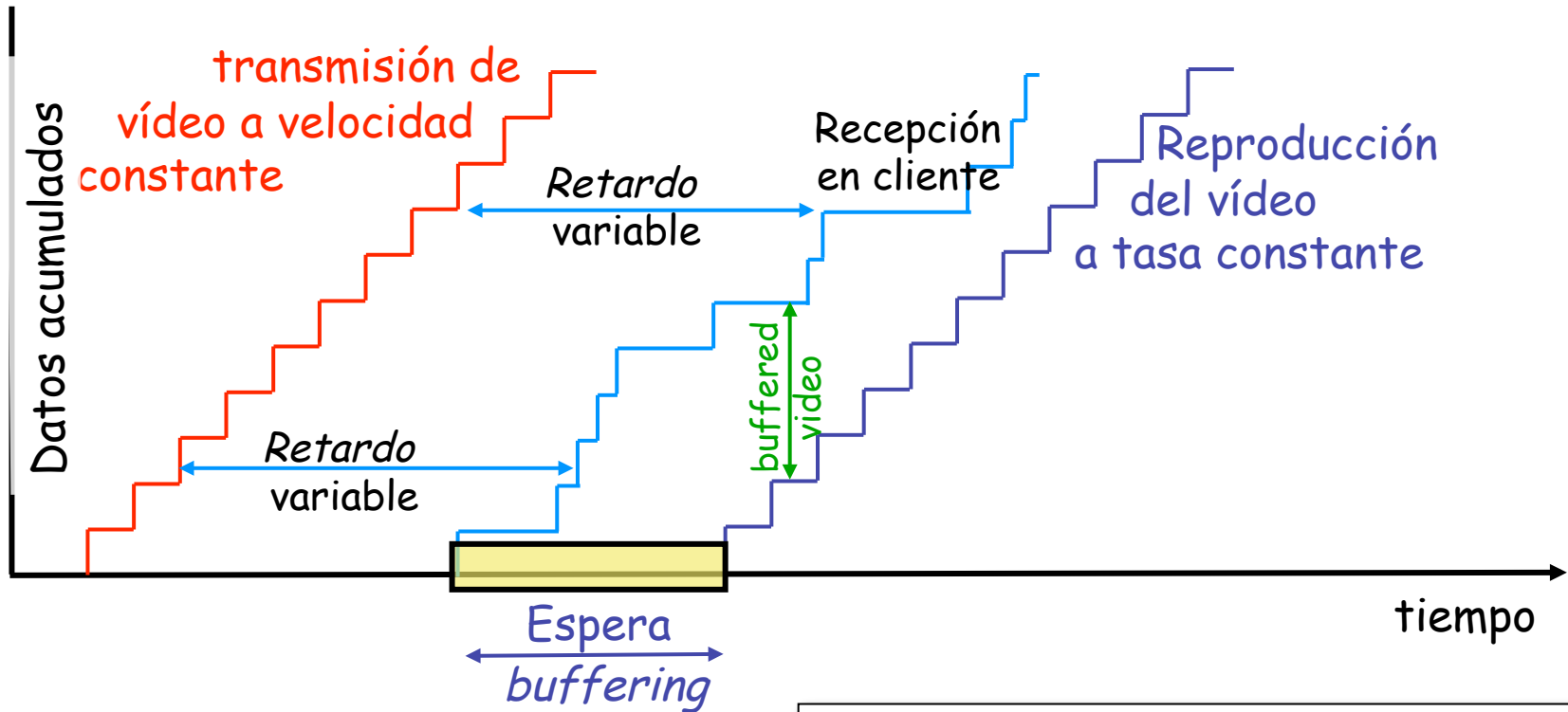


- Funcionalidades de VCR (vídeo):
 - Pausa, rebobinar, avance rápido, etc.
 - Retardo de comienzo 10 segs OK
 - Retardo ante un comando 1-2 seg OK
 - Protocolos para estos comandos (RTSP)

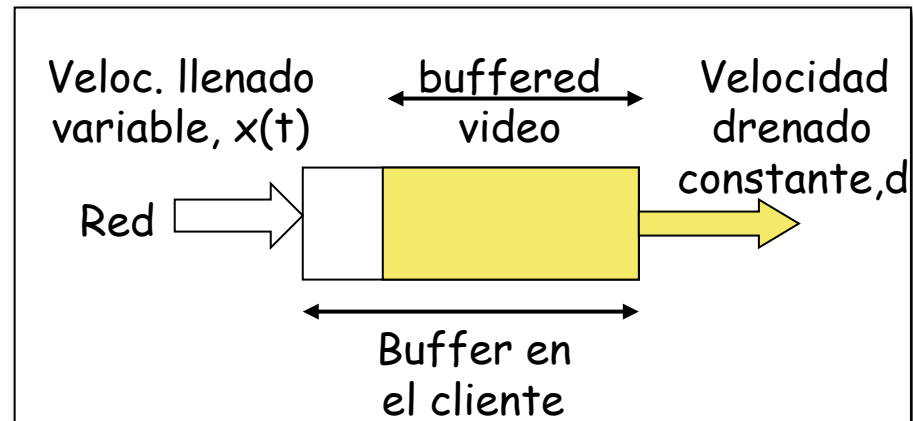
Streaming



Streaming: Client Buffering

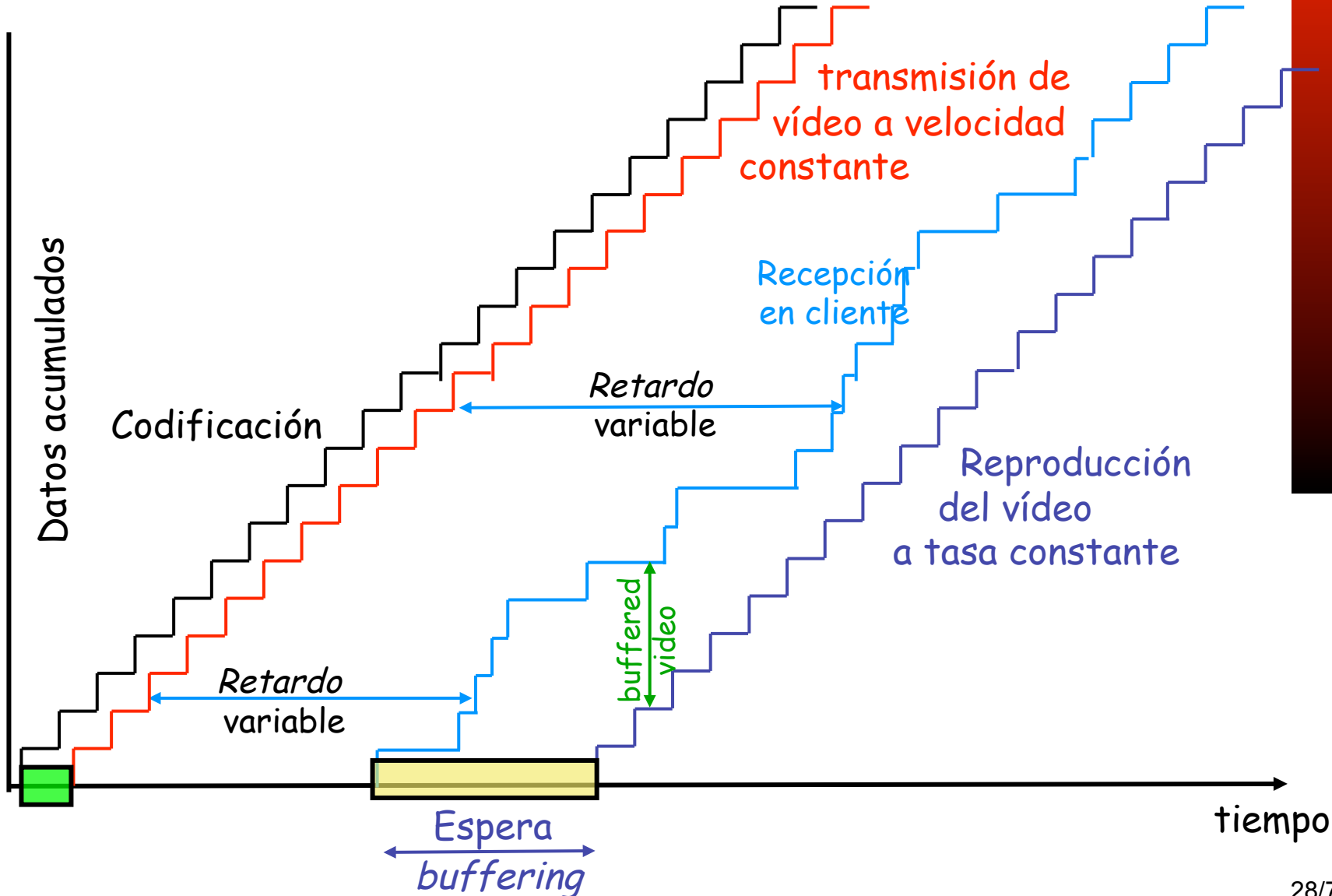


- El cliente acumula parte del vídeo en buffer antes de comenzar
- Permite absorber variaciones en el retardo en la red (*jitter*)
- Si el buffer se vacía el reproductor se detiene



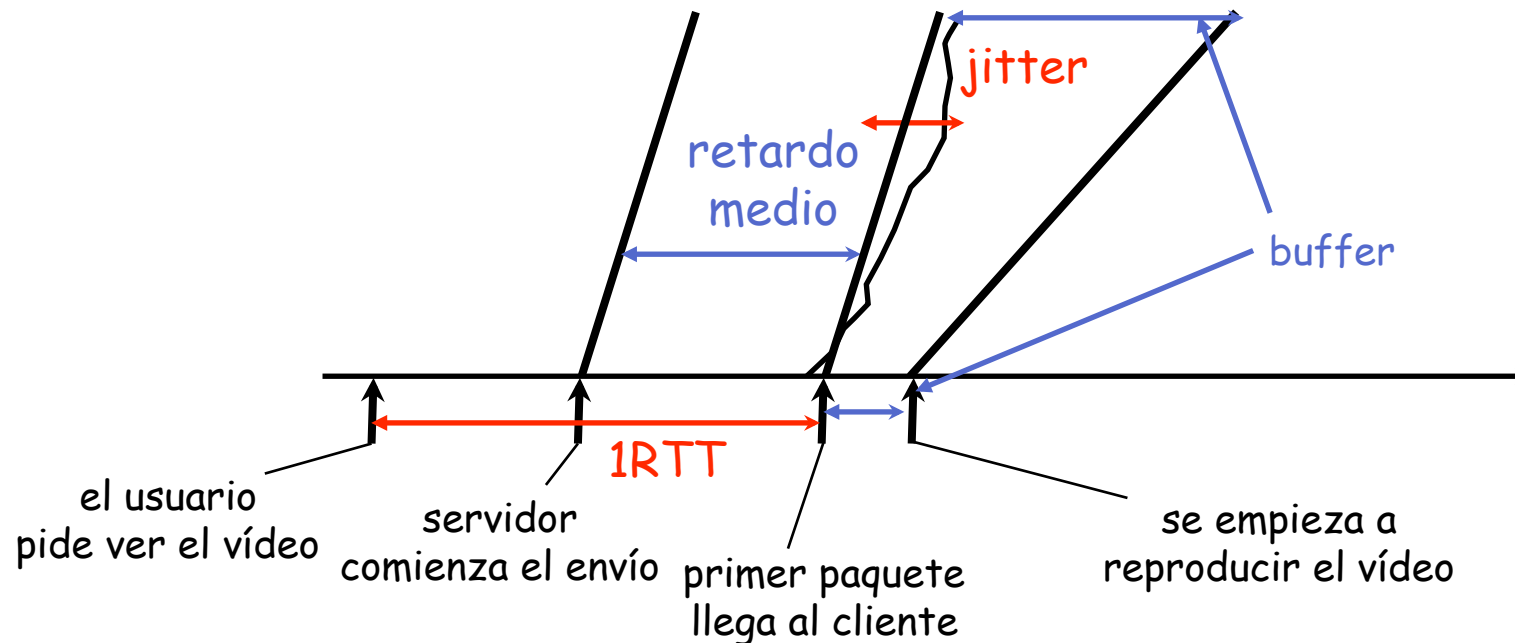
Live Streaming

- Avance rápido imposible



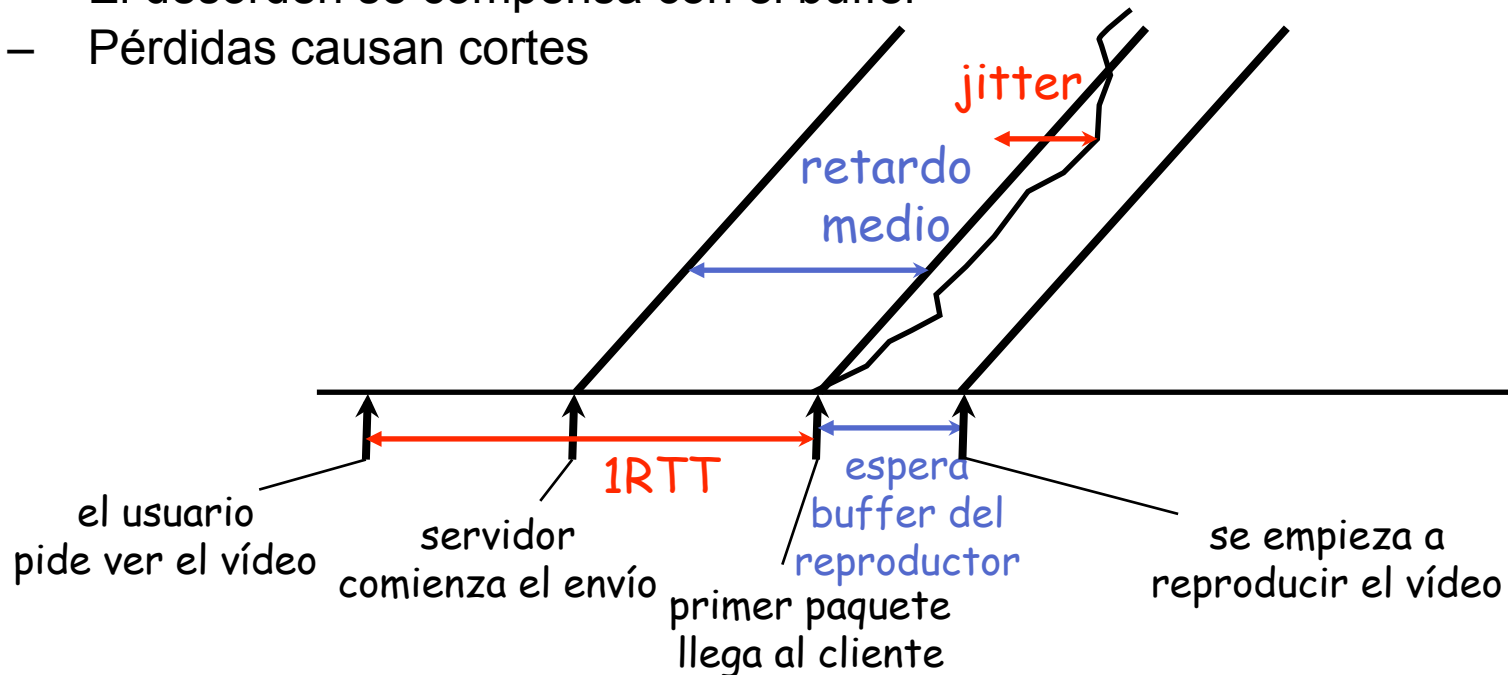
Requisitos de los servicios

- “Streaming” de fichero grabado (película)
 - Enviando a más velocidad que la de reproducción del vídeo
 - Disminuye la importancia del jitter
 - Necesita en general más buffer (un recurso)
 - Emplea más velocidad de la red (otro recurso)
 - Necesita que el vídeo esté disponible con antelación (no siempre es posible)



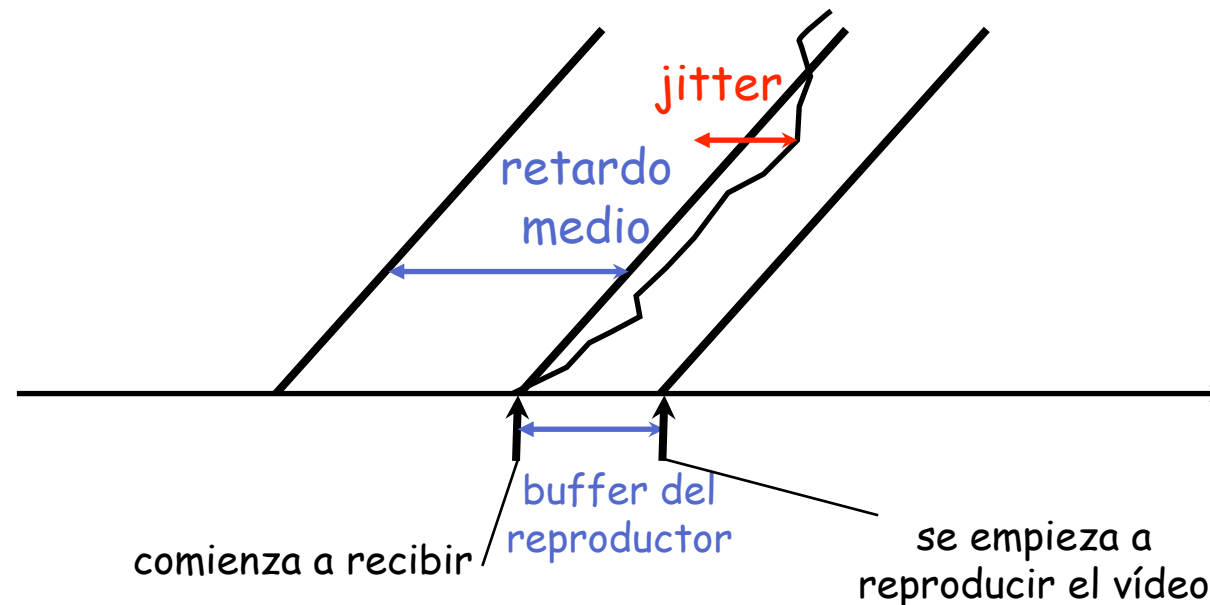
Requisitos de los servicios

- Streaming de fichero grabado (película)
 - BW suficiente para acomodar la tasa de la película
 - El tiempo de retardo one-way no importa demasiado
 - Contribuye al retardo inicial del comienzo de visualización
 - El jitter condiciona el tamaño del buffer
 - El buffer aumenta el retardo inicial del comienzo de visualización
 - Si el jitter es mayor que lo que puede absorber el buffer el vídeo puede detenerse
 - El desorden se compensa con el buffer
 - Pérdidas causan cortes



Requisitos de los servicios

- Streaming en vivo (no podemos tener el vídeo a más velocidad de la que se genera)
- O un canal de TV programación lineal (no podemos tener buffer para todo el vídeo)
 - BW suficiente para acomodar la tasa de la película
 - Retardo irrelevante
 - Desorden irrelevante



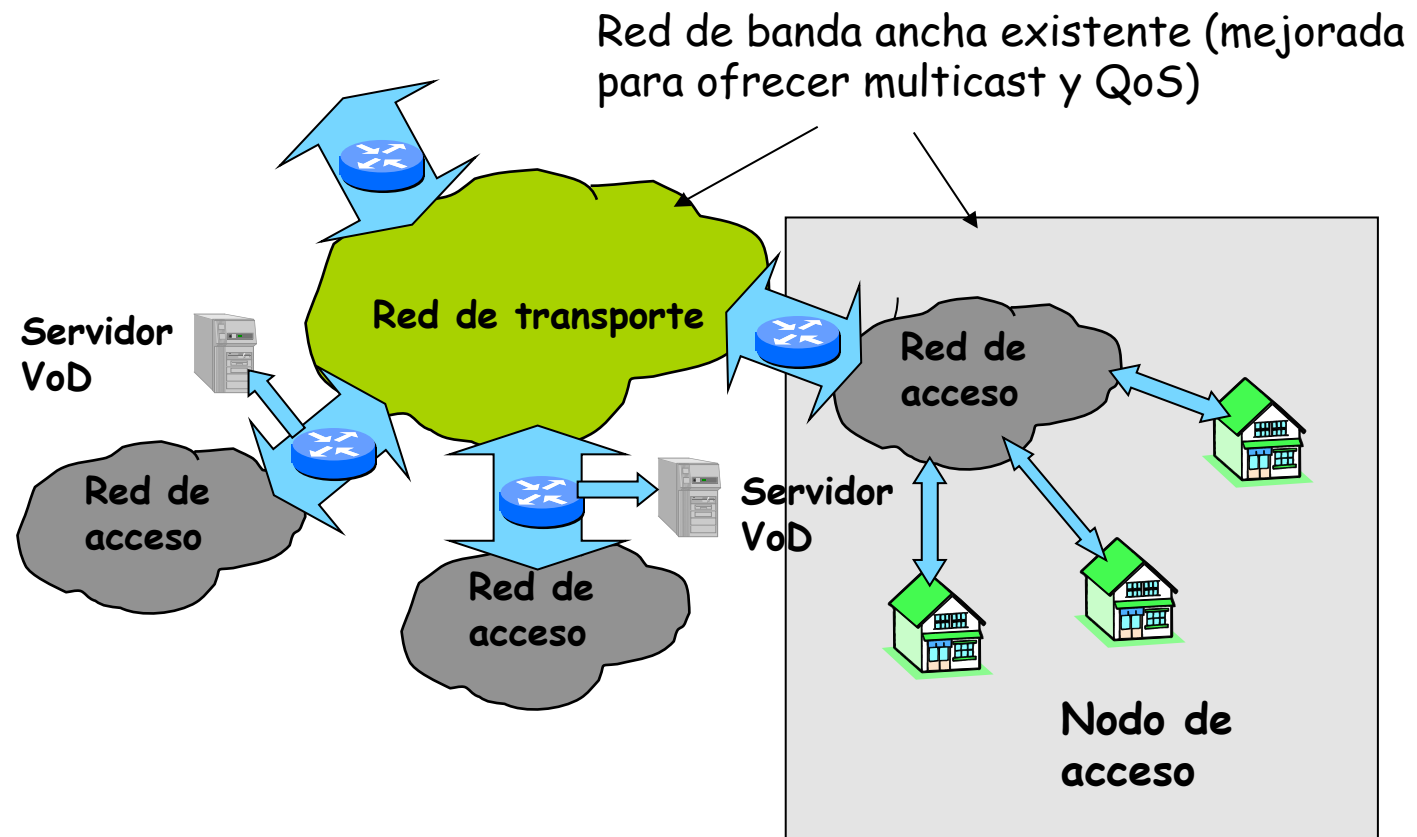
Arquitectura del servicio

Necesidades de la red

- BW
- Retardo (velocidad de interacción)
- Jitter (frecuencia de cortes o tamaño del buffer)
- Pérdidas (frecuencia de cortes)

- Situación actual
 - Vídeo sobre internet. Sobredimensionar BW y esperar que los otros tres no sean demasiado elevados, sin garantías
 - Vídeo en entornos controlados por un operador
 - IPTV
 - VoD
 - Mezclas, batching

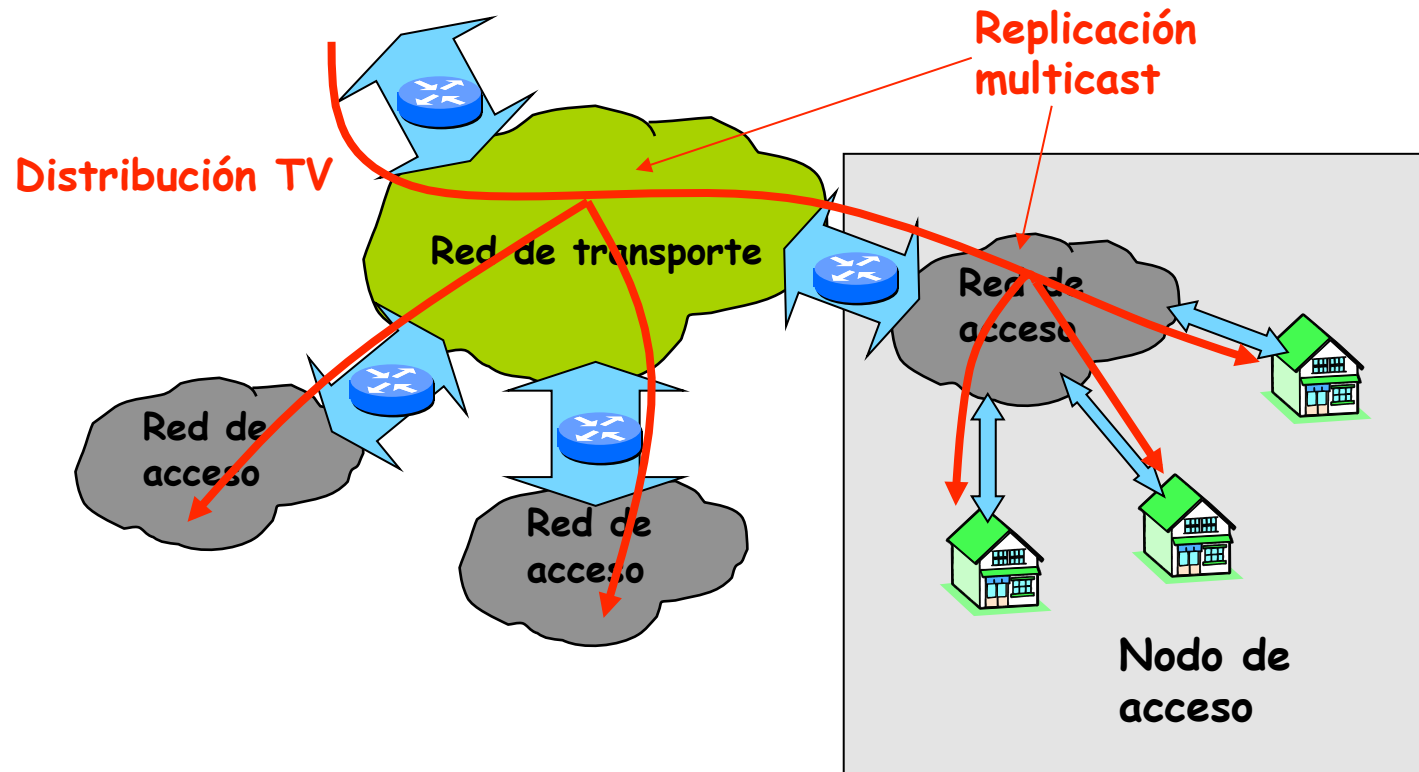
Arquitectura general



Difusión de TV

Multicast para difusión de TV

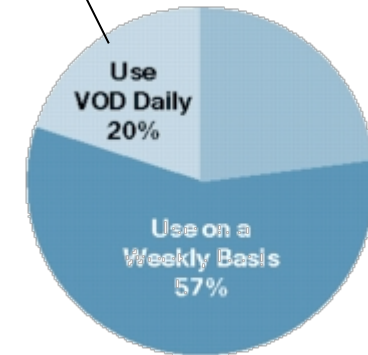
- Escalabilidad para un alto número de canales y un número ilimitado de usuarios
- Uso óptimo del ancho de banda



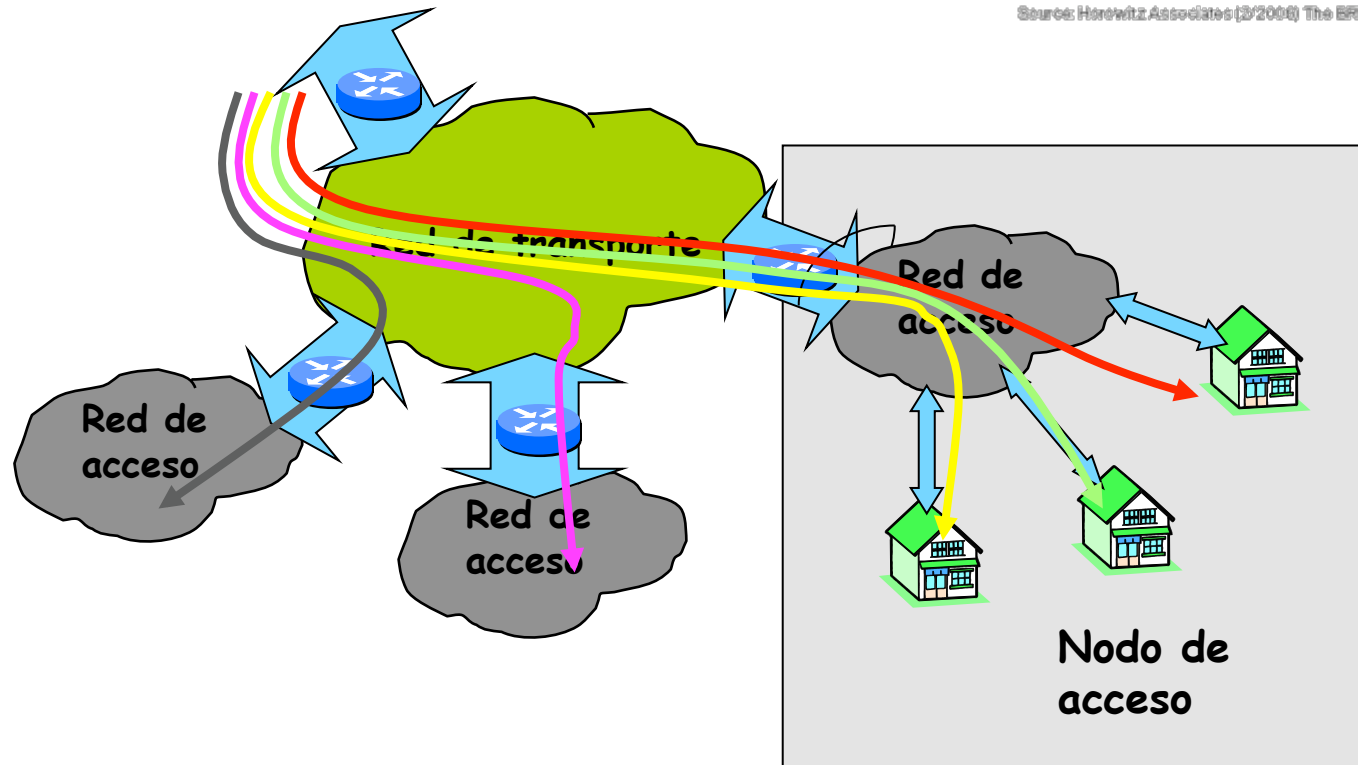
VoD

VoD equivale a Unicast

- 1M usuarios → 200K/día @ 2Mb/s & 20% concurrencia = 80Gb/s
- Arquitectura centralizada proporciona escalabilidad limitada



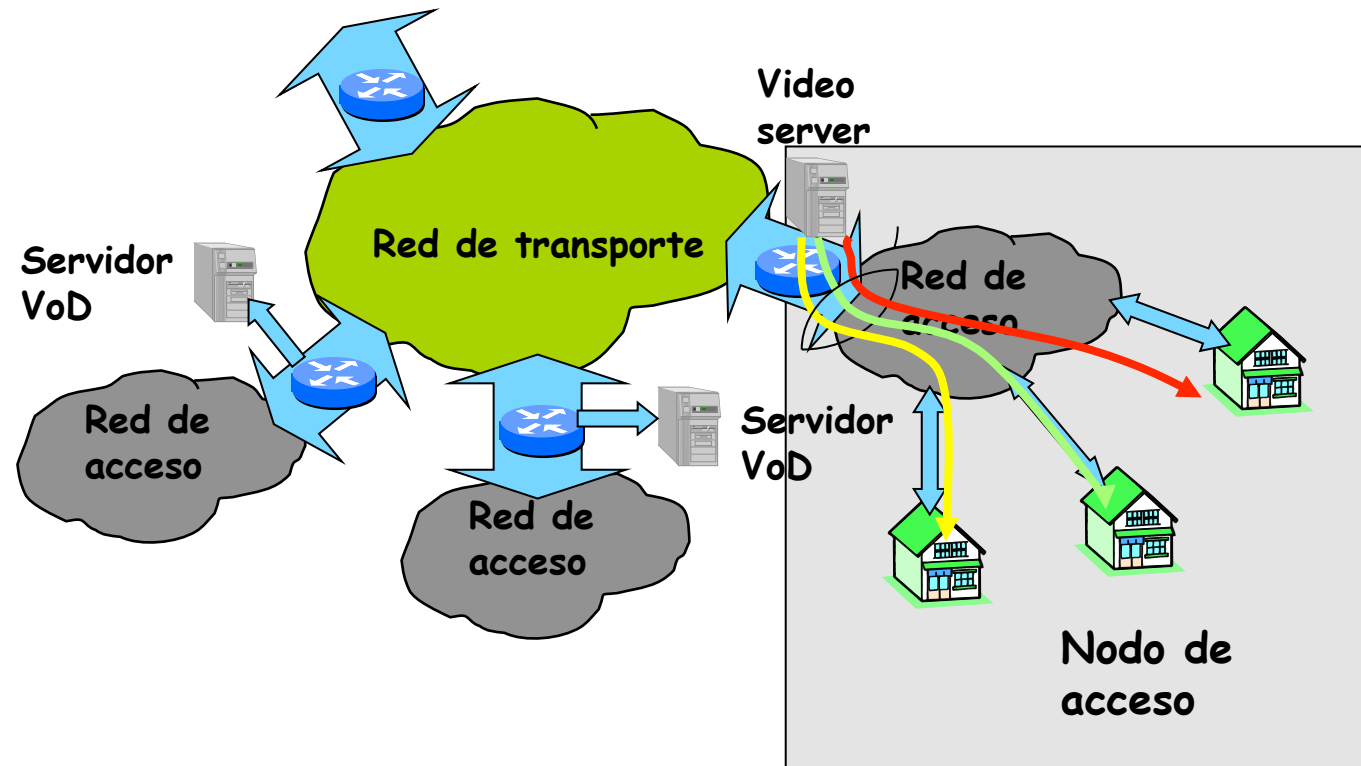
Source: Horowitz Associates (3/2006) The BRIDGE 2006



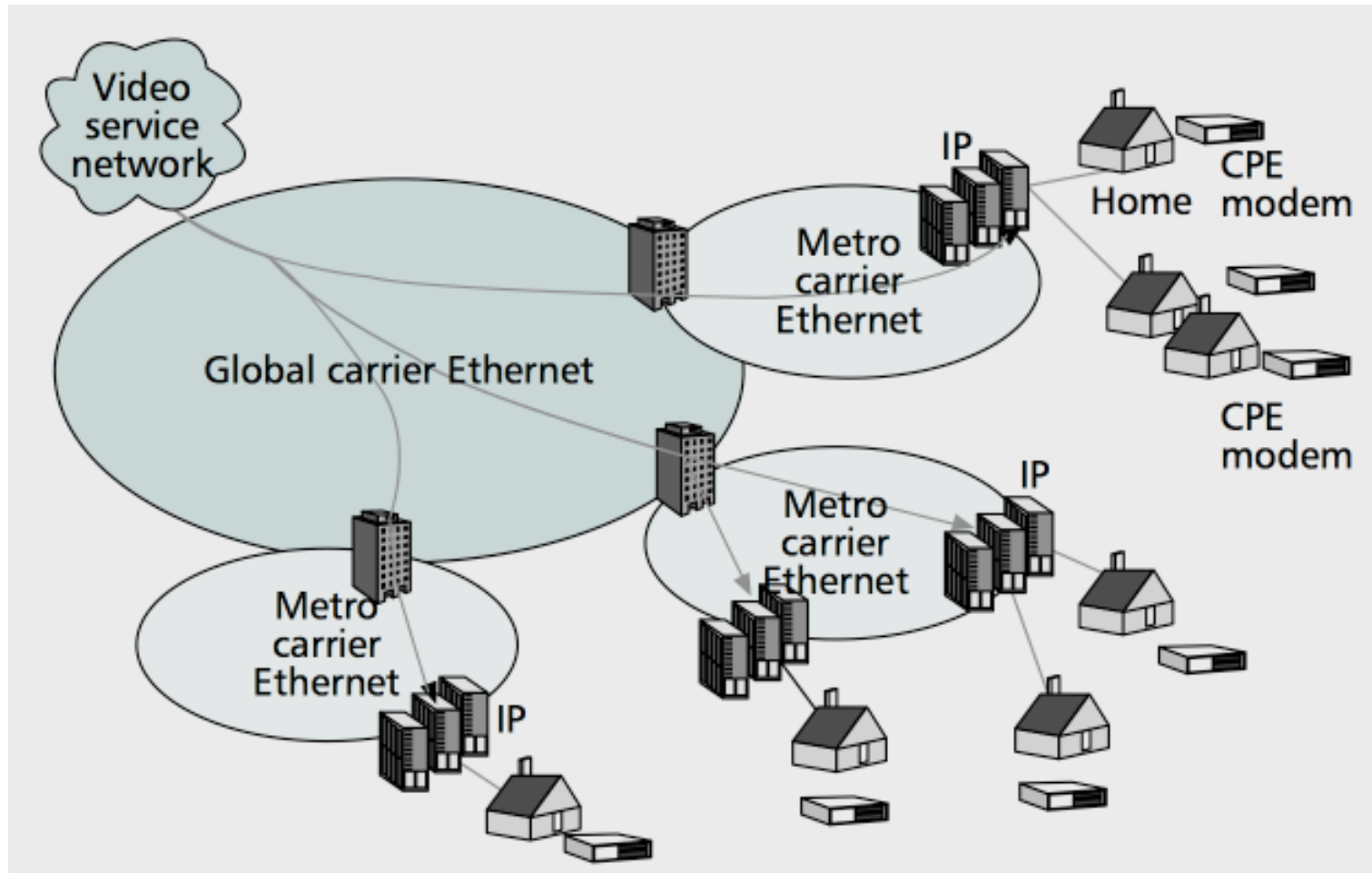
VoD

Arquitectura distribuida de VoD

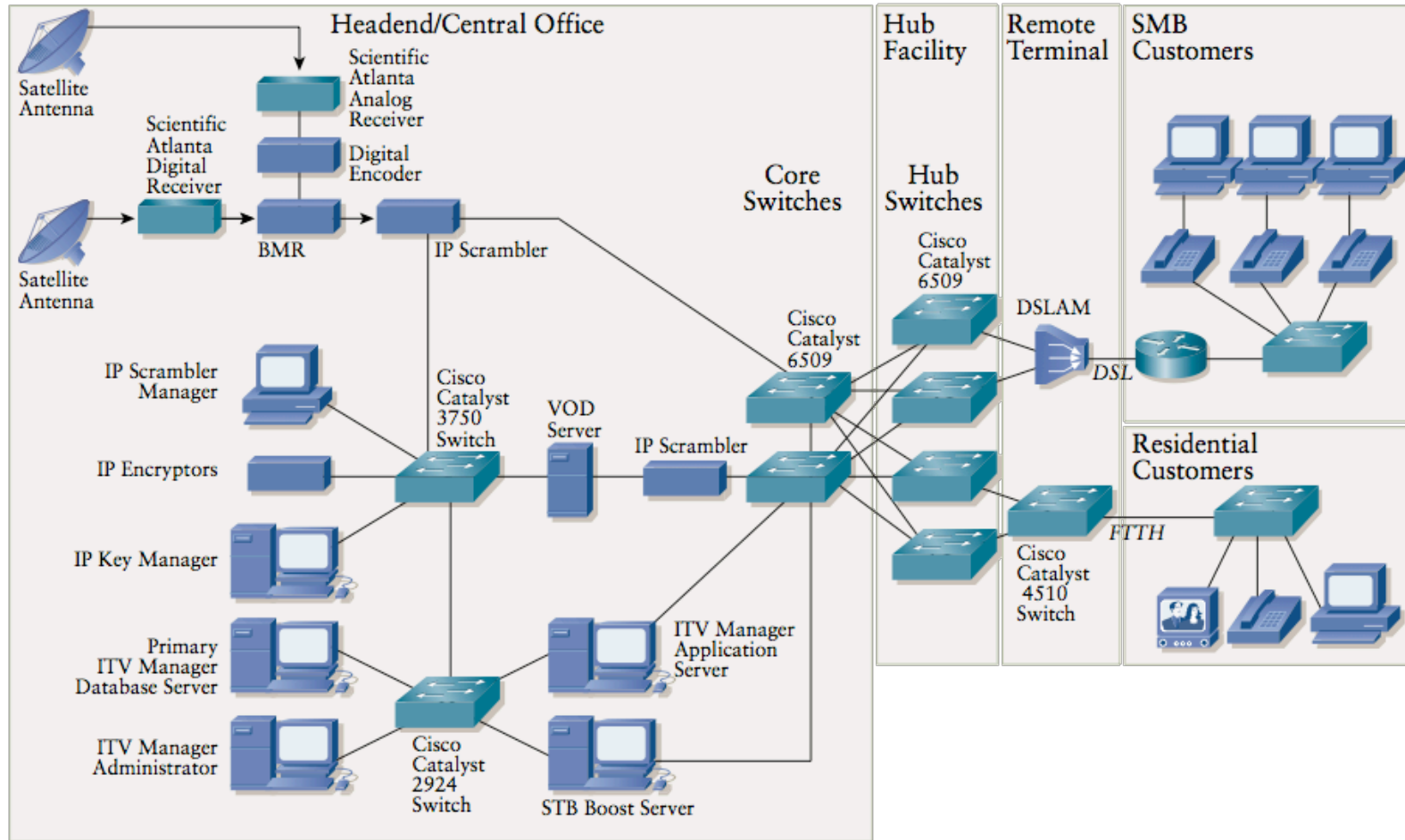
- 20k usuarios @ 2Mb/s & 20% concurrencia = 8Gb/s
- Arquitectura distribuida proporciona escalabilidad
- También es posible una arquitectura mixta (centralizada/distribuida)



Arquitecturas de red

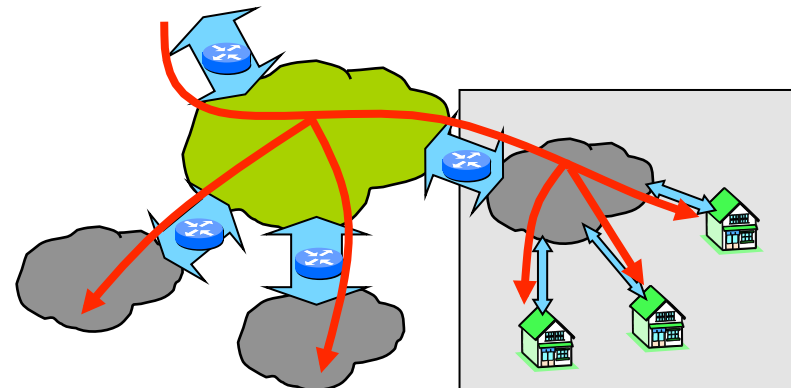


Arquitecturas de red



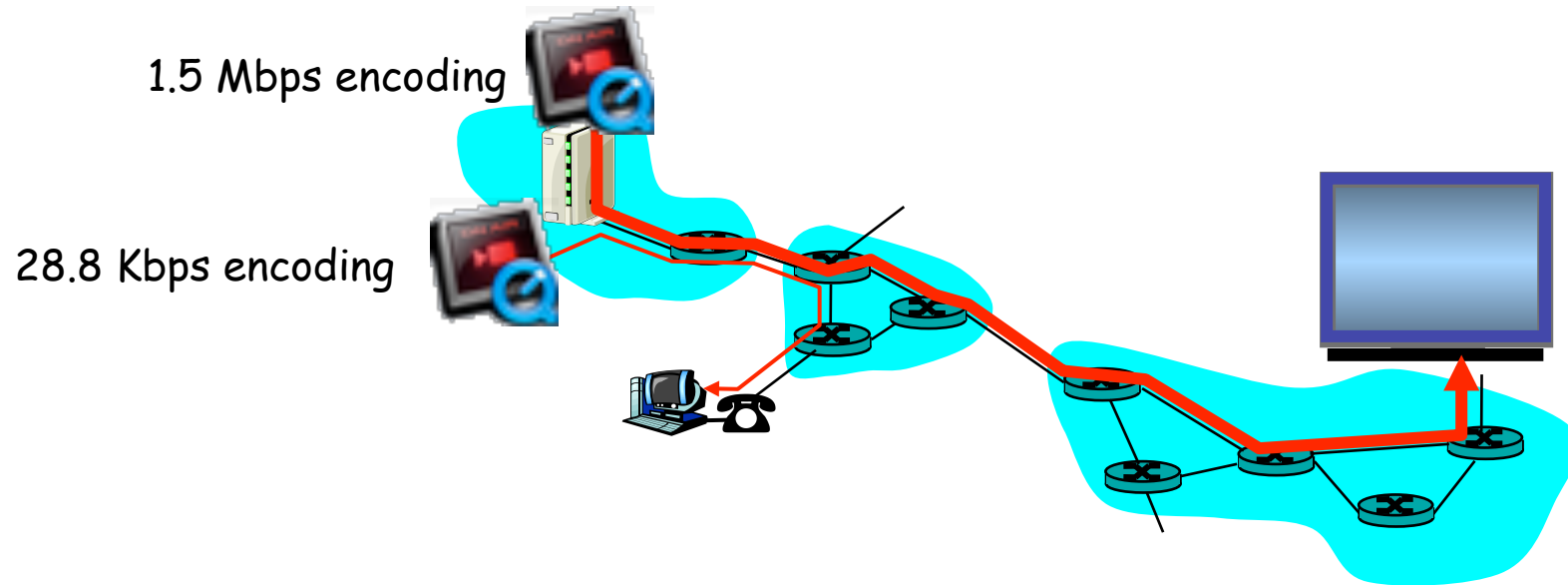
¿Necesidades?

- Servidor:
 - Escalable ante numerosas peticiones
- Red:
 - Escalabilidad \Rightarrow Multicast
 - Sin cortes \Rightarrow QoS (tecnología?)
- Dimensionamiento:
 - ¿Cómo es el tráfico de vídeo?
 - ¿Cómo se comportan los usuarios?



El servidor

Streaming: client rate(s)



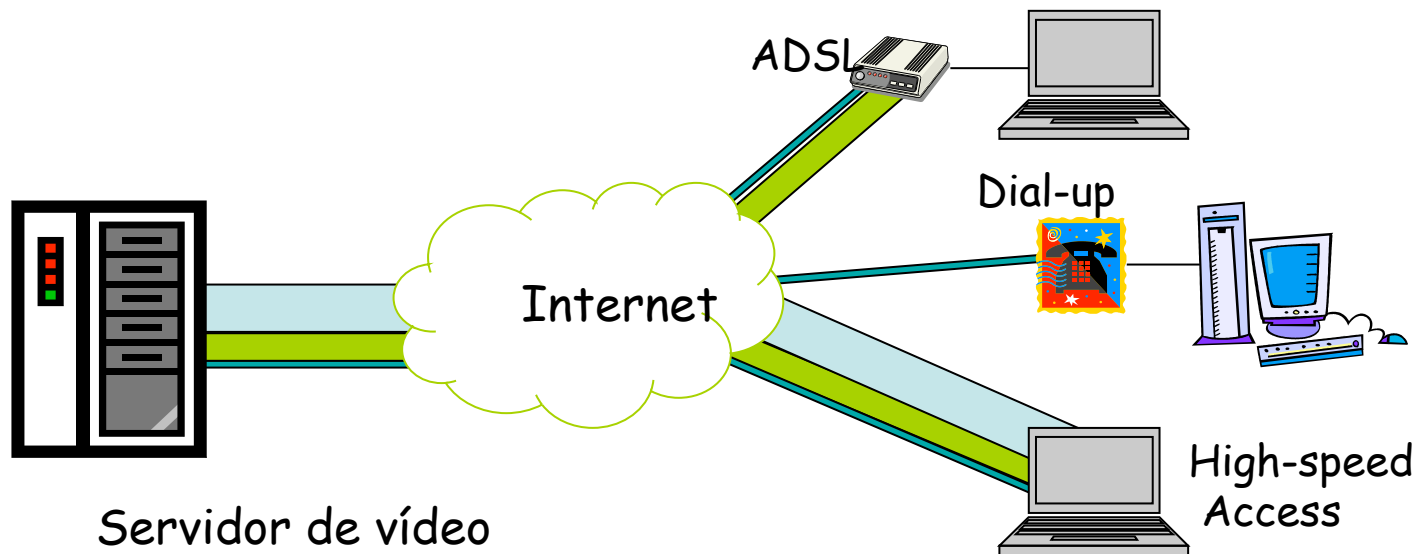
Q: ¿Cómo gestionar diferentes capacidades de recepción?

- 28.8 Kbps modem
- 100Mbps Ethernet

A: El servidor almacena y transmite copias del vídeo codificadas a diferentes velocidades

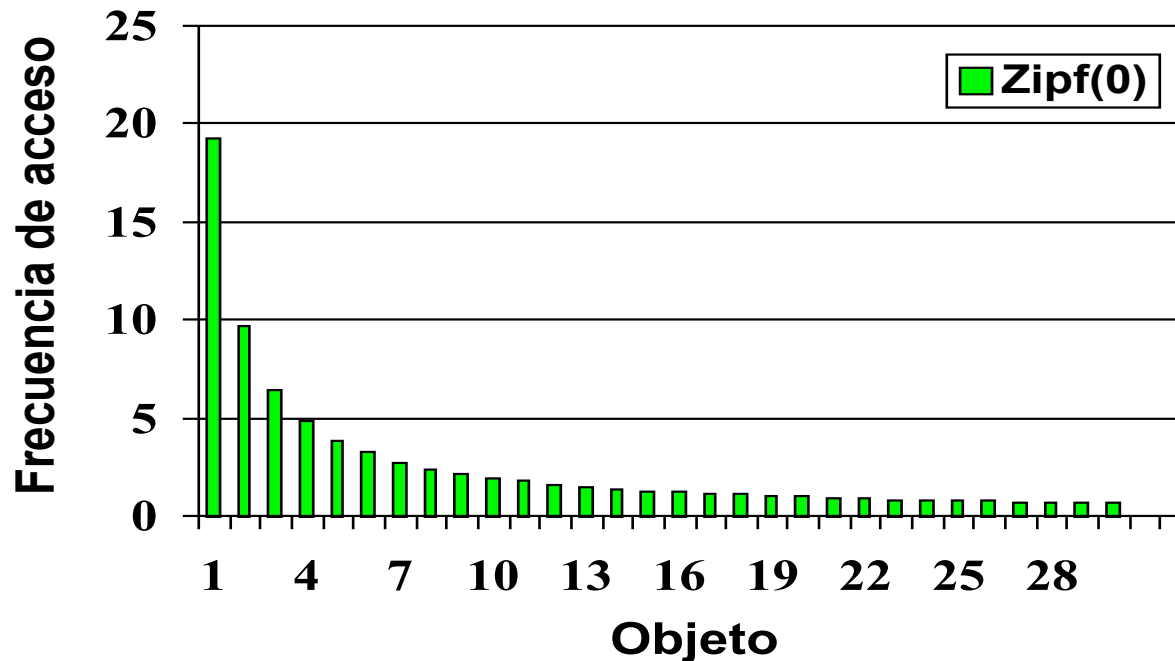
Streaming Live Multimedia

- ¿Cómo hacer *streaming* a un gran número de clientes?
 - Ejemplo: Un evento deportivo popular
 - Usar multicast/broadcast
- ¿Y la heterogeneidad de clientes?
 - Los clientes pueden disponer de diferente BW
 - Vídeo en capas



Comportamiento de los usuarios

- 100s – 1000s solicitudes de un fichero / duración de su reproducción
- Popularidad sesgada de los ficheros
 - 10% – 20% de los ficheros acaparan el 80% de las peticiones
 - Tiene sentido centrarse en los ficheros populares



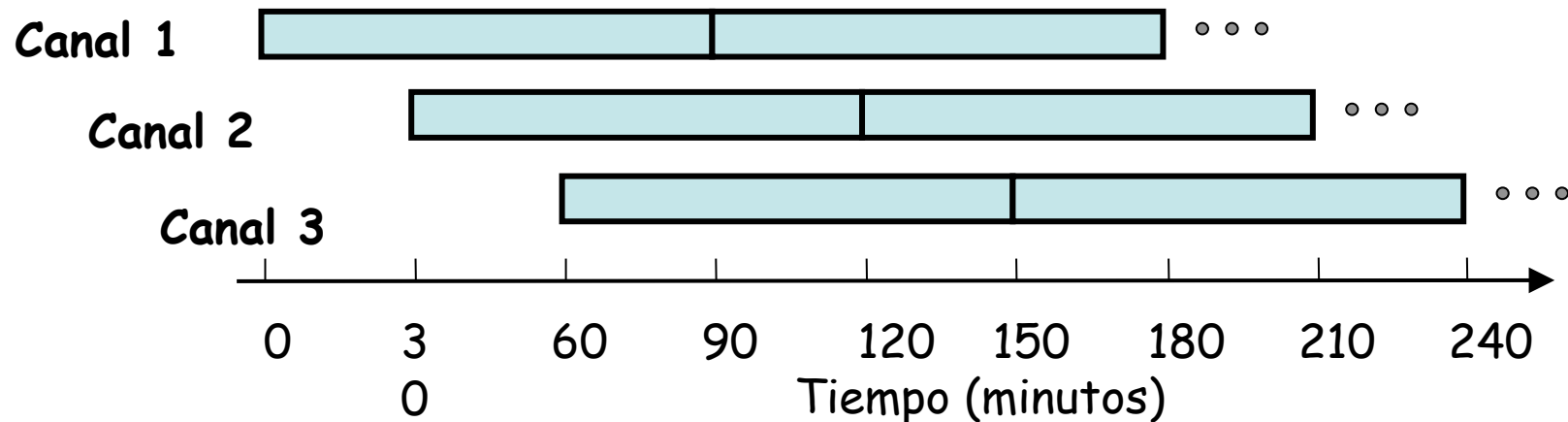
Streaming escalable: Motivación

- Consideremos un fichero popular
 - Tasa de reproducción (*playback rate*): 1 Mbps
 - Duración: 90 minutos
 - Peticiones: una por minuto
- Comienza un nuevo stream por cada petición
 - BW requerido = 1 Mbps x 90
- ¿Cómo dar flexibilidad en el instante de comienzo sin enviar un flujo por petición?
- *Batching*
 - Acumular suficientes peticiones para que sea rentable iniciar un nuevo flujo
 - Empezar el flujo al acumular suficientes usuarios o llegar a un límite de tiempo de espera



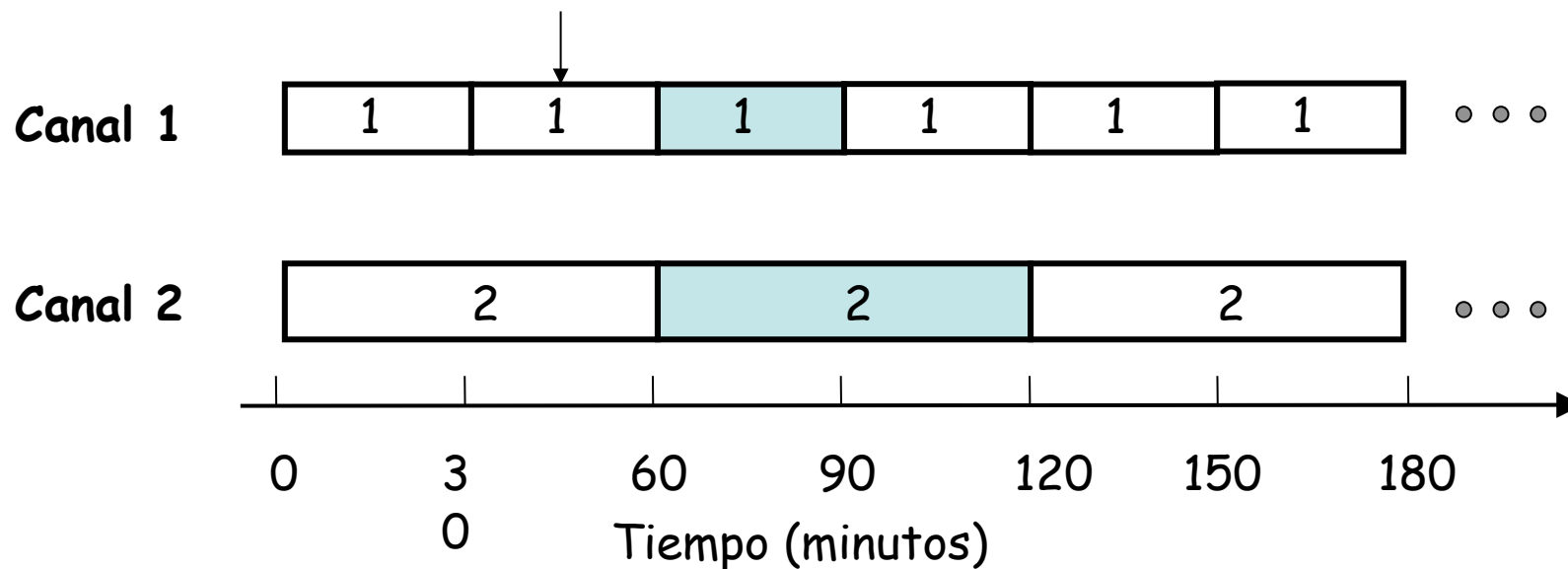
Batching (Ejemplo)

- Playback rate = 1 Mbps, duración = 90 minutos
- Agrupar peticiones en intervalos no solapados de 30 minutos:
 - Máx espera comienzo = 30 minutos
 - BW necesario = 3 canales = 3 Mbps
- BW aumenta linealmente con la reducción de la espera de comienzo



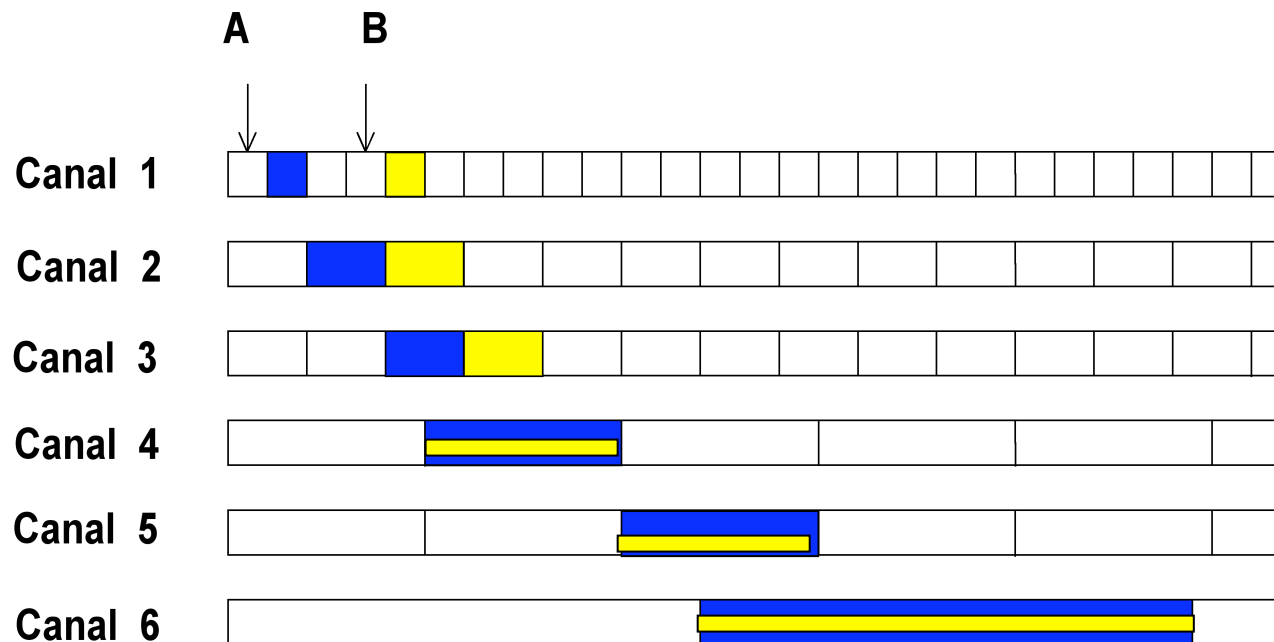
Periodic Broadcast

- Partir el fichero en 2 segmentos con tamaños relativos {1,2}
- Para una película de 90 minutos:
 - Segmento 1 = 30 minutos, Segmento 2 = 60 minutos
- Ventaja:
 - Máximo tiempo de espera comienzo = 30 minutos
 - BW necesario = 2 canales = 2 Mbps
- Desventajas: Requiere más del cliente (recibir 2 a la vez)



Skyscraper Broadcasts (SB)

- Divide el fichero en K segmentos de tamaños crecientes
 - Progresión de tamaños: 1, 2, 2, 5, 5, 12, 12, 25, ...
- Multicast de cada segmento en un canal separado
- Tasa a cada cliente: 2 x playback rate



Streaming: ¿UDP o TCP?

UDP

- El servidor envía a la velocidad apropiada para la reproducción
- Retardo de comienzo (2-5 segs) para compensar el jitter de la red
- Recuperación de pérdidas: *time permitting*

TCP

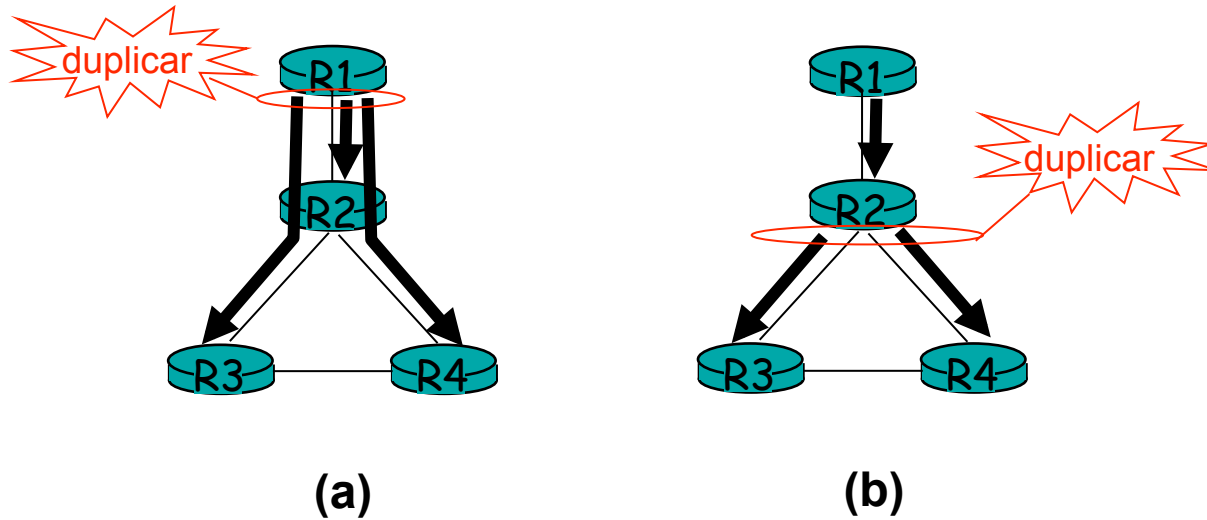
- Puedes entregar a la velocidad de reproducción pero no sabes cuándo lo enviará TCP
- Además puede acumular bytes para formar paquetes más grandes
- Normalmente envía a la máxima velocidad
- Fluctuará debido a los mecanismos de control de flujo y control de congestión
- Retardo de comienzo más largo para poder suavizar el comportamiento de TCP
- HTTP/TCP atraviesa mejor firewalls

Multimedia en la Internet actual

- TCP/UDP/IP: “best-effort service”
- Sin garantías de retardo o pérdidas
- Los programas emplean técnicas en el nivel de aplicación para mitigar los efectos de retardos y pérdidas
 - Buffers
 - Algoritmos de codificación resistentes ante pérdidas

Necesidades en la red: Multicast

Multicast: Introducción



Duplicar en el origen frente a duplicar en la red
(a) En el origen, (b) en la red

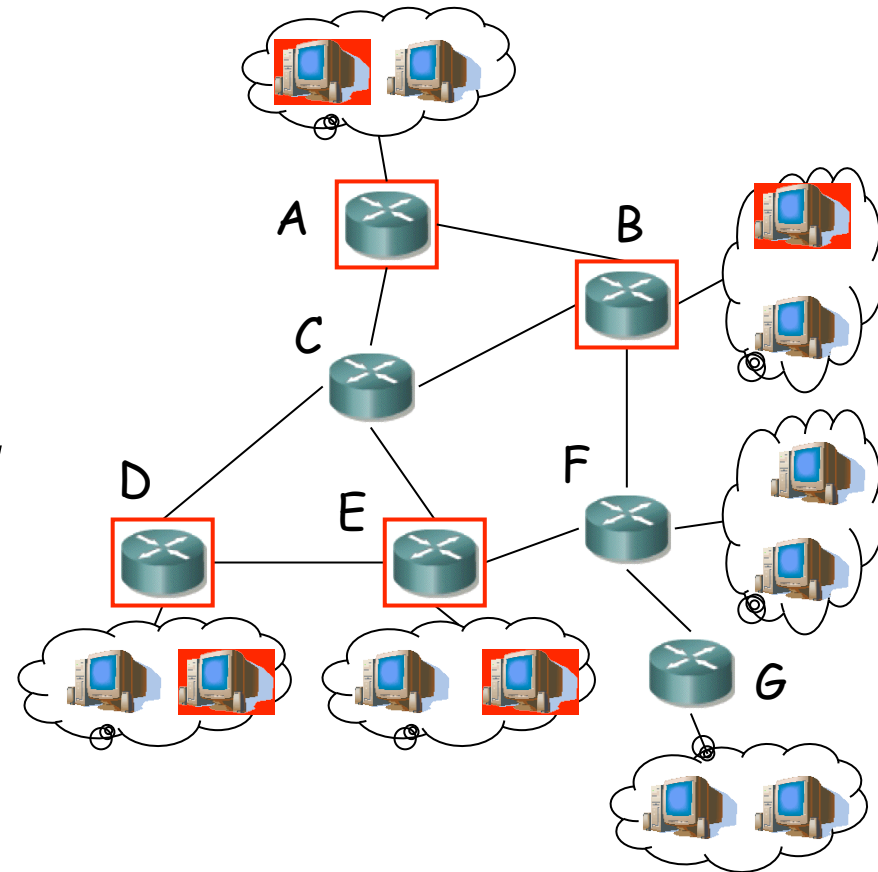
Multicast

- Grupo abierto
 - Grupo identificado por una dirección independiente de la localización
 - Cualquiera (incluso de fuera del grupo) puede enviar a los miembros del grupo
- Sencillo dentro de LANs que soportan broadcast
- Complicado atravesando redes

¿Cómo llegan los datagramas a todos los miembros del grupo?

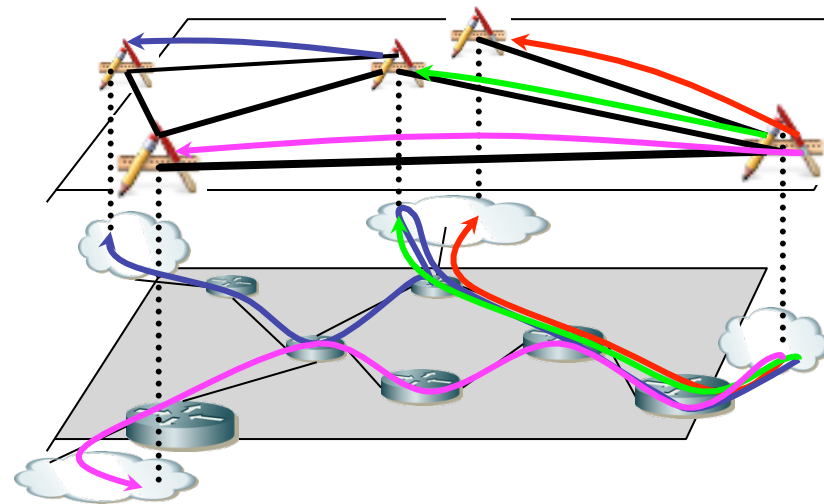
Multicast Routing

- Encontrar un árbol de enlaces que conecte a todos los routers que sirven a hosts del grupo
- Árboles de expansión (*spanning trees*)
- Mínimos respecto a una métrica (*minimum spanning trees*)
- Puede implicar a otros routers



Application layer multicast

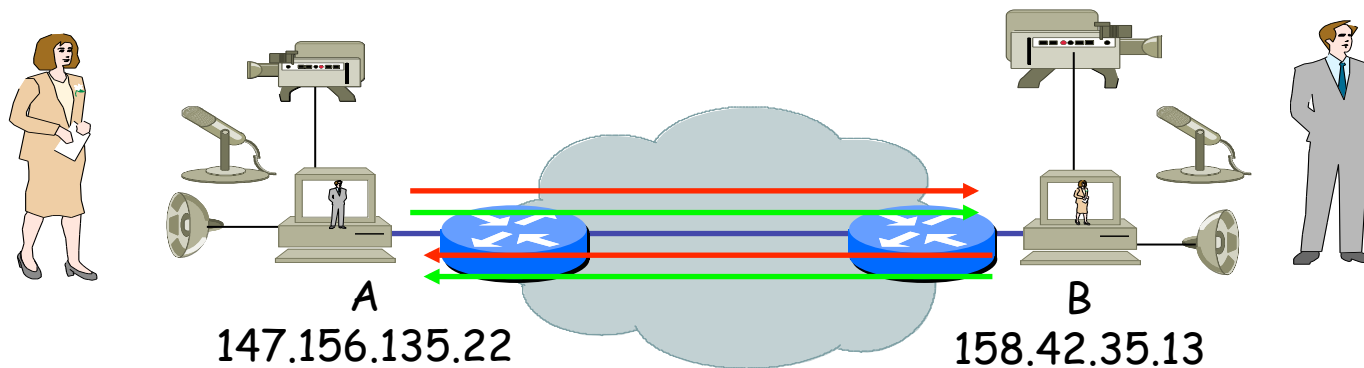
- Multicast en el nivel de aplicación
 - Asumiendo que la red subyacente ofrece solo facilidades unicast
 - Los sistemas finales (aplicaciones en los hosts) se comunican a través de una *overlay*
- Ventajas
 - No hay que cambiar los routers
 - Permite incorporar con facilidad nuevas funcionalidades
- Problemas
 - ¿Cómo pueden cooperar para construir una buena estructura overlay?
 - Serias implicaciones en las prestaciones por no optimizar tráfico



Necesidades en la red: QoS

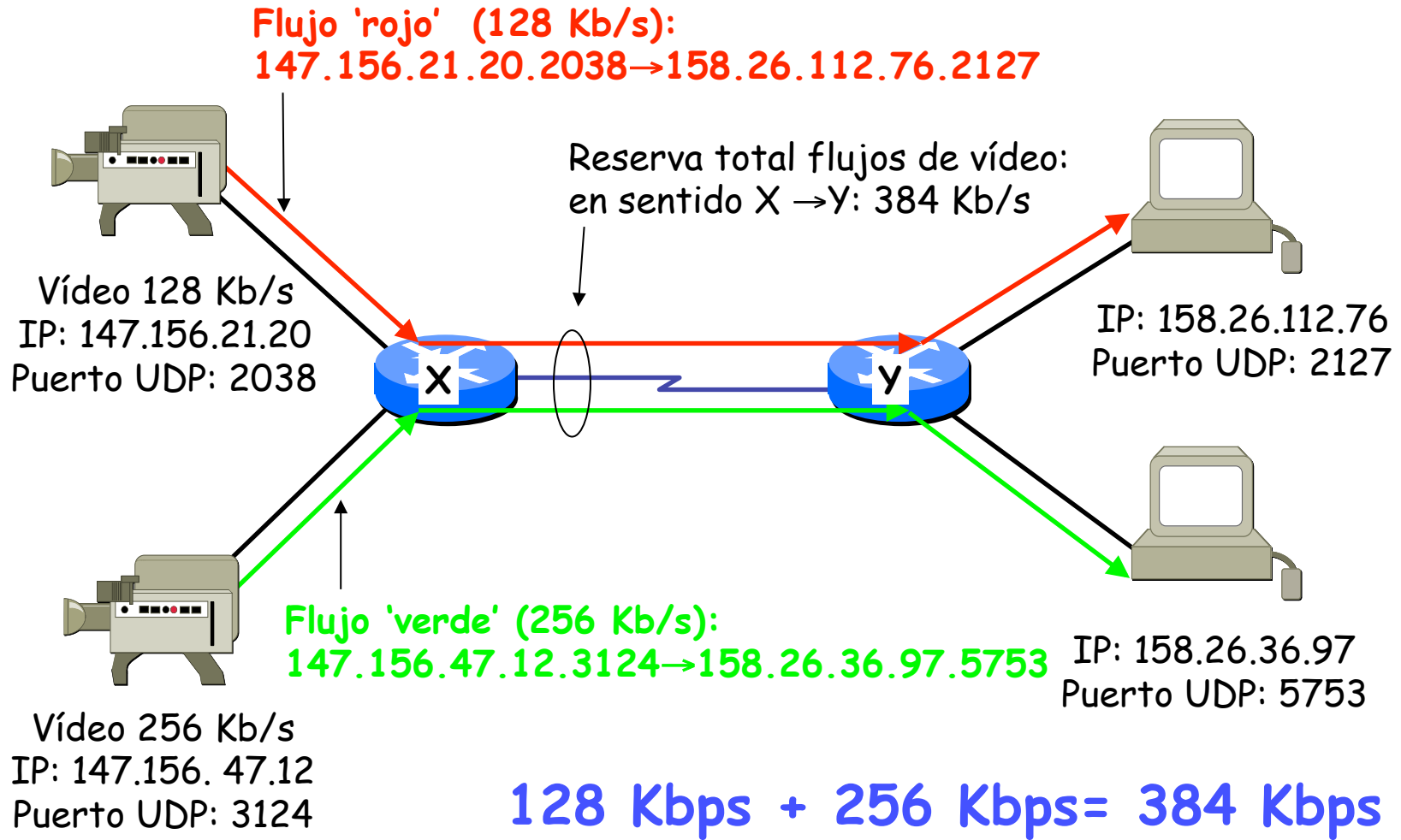
Concepto de flujo en QoS

- Secuencia de datagramas que se produce como resultado de una acción del usuario y requiere la misma QoS
- Normalmente es simplex (unidireccional)
- Es la entidad más pequeña a la que los routers pueden aplicar una determinada QoS
- Ejemplo: una videoconferencia estaría formada por cuatro flujos, dos en cada sentido, uno para el audio y otro para el vídeo.
- Los flujos pueden agruparse en clases; todos los flujos dentro de una misma clase reciben la misma QoS.



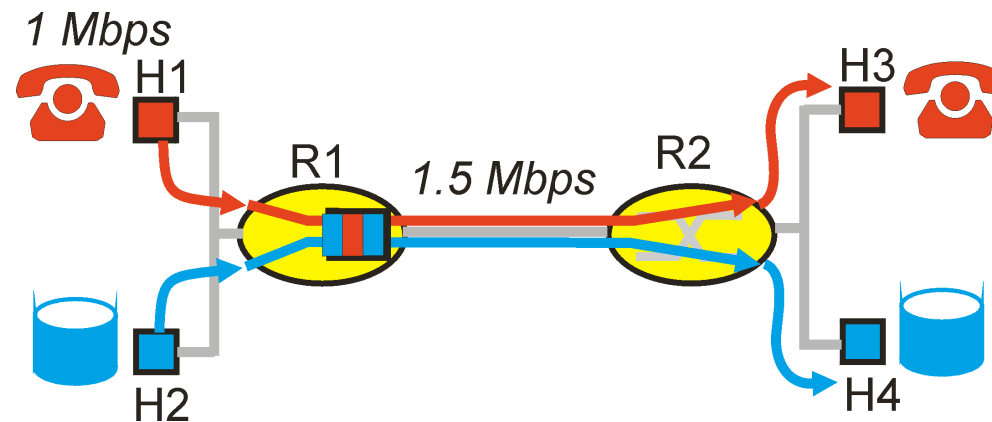
- Flujo vídeo A->B: 147.156.135.22:2056 -> 158.42.35.13:4065
- Flujo audio A->B: 147.156.135.22:3567 -> 158.42.35.13:2843
- ← Flujo vídeo B->A: 158.42.35.13:1734 -> 147.156.135.22:6846
- ← Flujo vídeo B->A: 158.42.35.13:2492 -> 147.156.135.22:5387

Agrupación de flujos o clases en vídeo



Principio 1: Clasificación

- Ejemplo: Teléfono IP a 1Mbps, comparte enlace de 1.5Mbps con FTP
 - Ráfagas de FTP pueden congestionar el router y causar fallos en el audio
 - Queremos dar prioridad al audio sobre el FTP

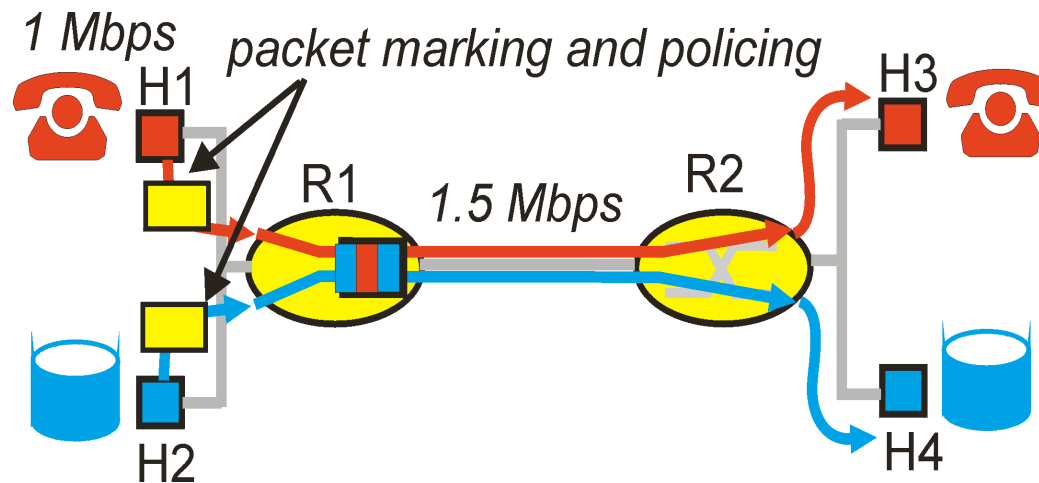


Principio 1

Los routers necesitan distinguir el tráfico de diferentes clases y aplicarles diferentes políticas: *packet marking* (generalmente a la entrada a la red)

Principio 2: Aislamiento

- ¿Qué sucede si las aplicaciones no se comportan como deben?
 - Por ejemplo la aplicación de audio envía más de lo previsto
 - Necesitamos forzar que las fuentes se comporten como se ha acordado

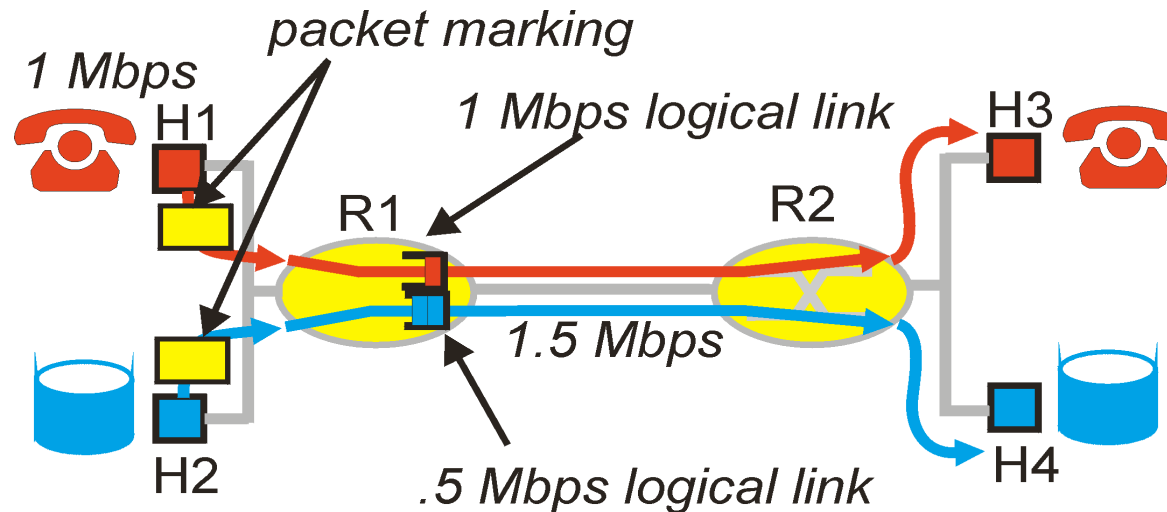


Principio 2

Forzar que una clase de tráfico se comporte dentro de lo contratado: *policing* (típicamente a la entrada)

Principio 3: Eficiencia

- Reservar BW fijo (no compartido) a un flujo es ineficiente si ese flujo no lo emplea todo

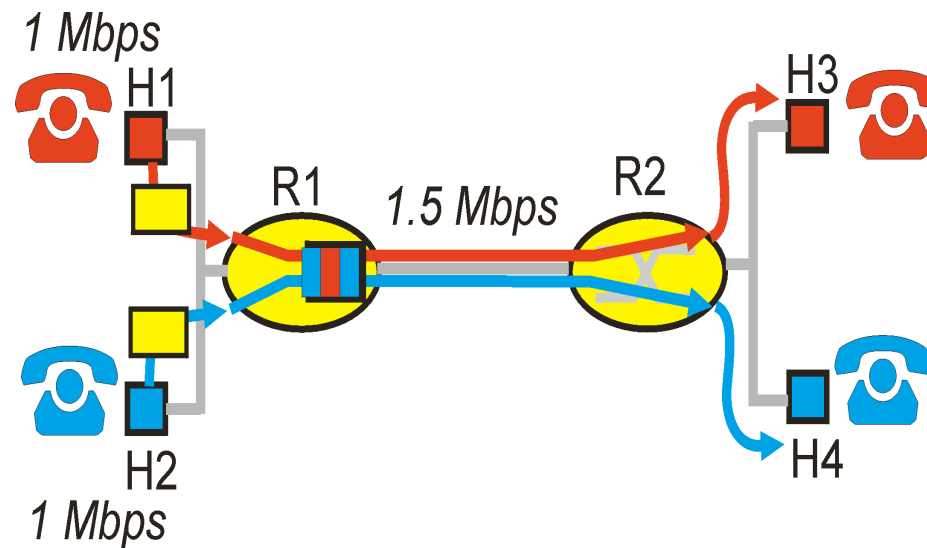


Principio 3

Mientras se ofrece aislamiento es deseable emplear los recursos de forma eficiente (*work conserving*): *scheduling* (en todos los routers del camino)

Principio 4: Límite de recursos

- No se pueden satisfacer las demandas más allá de la capacidad del enlace

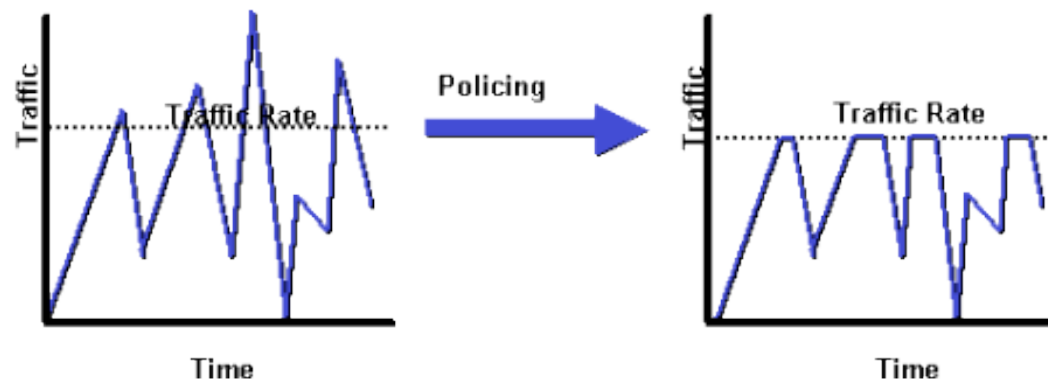


Principio 4

El flujo declara sus necesidades pero la red puede *bloquear* al flujo si no puede satisfacerlas: *call admission*

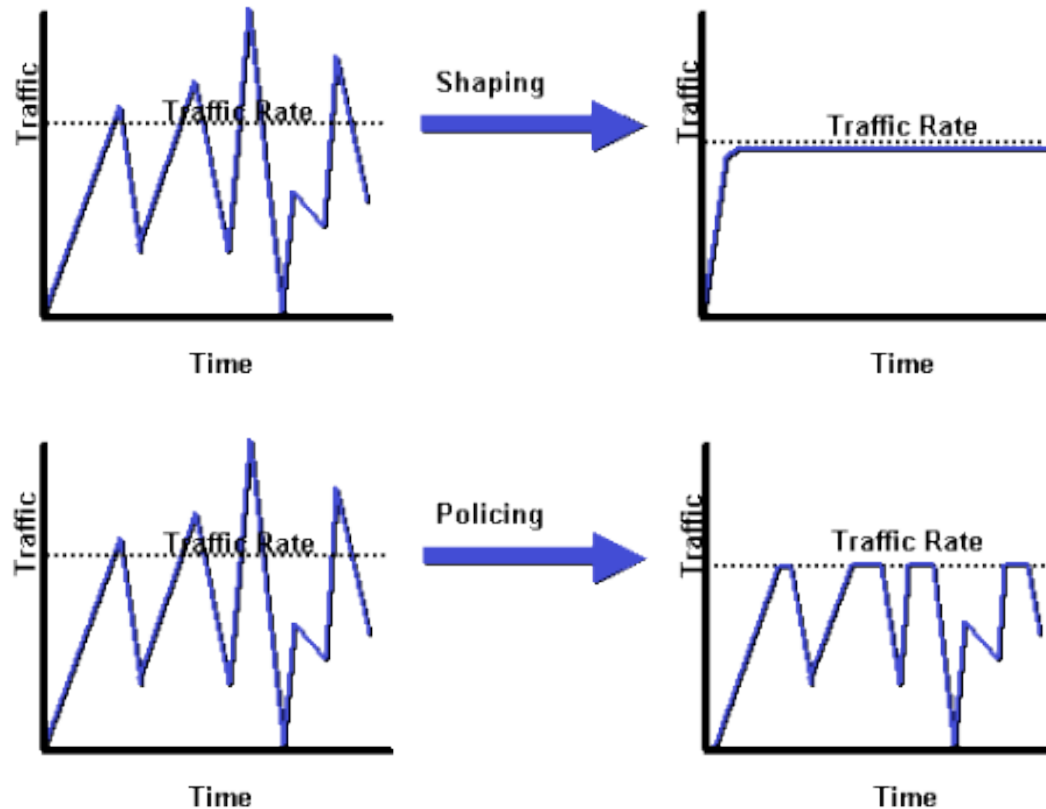
Policing

- **Objetivo:** Limitar el tráfico a la entrada a la red para que no exceda el declarado
- Características del tráfico
 - Tasa media (media a largo plazo)
 - Tasa de pico
 - Tamaño máximo de ráfaga: máx nº paquetes a tasa de pico
- Los que excedan se descartan o marcan (*conditional marker*)
- Policer as Marker:
 - Puede que el tráfico ya venga marcado según la aplicación a la que pertenezca: Separación vertical del tráfico
 - Un *policer* no distingue la aplicación, marca en función del tráfico: Separación horizontal del tráfico



Shaping

- Los que excedan no se descartan sino que se encolan

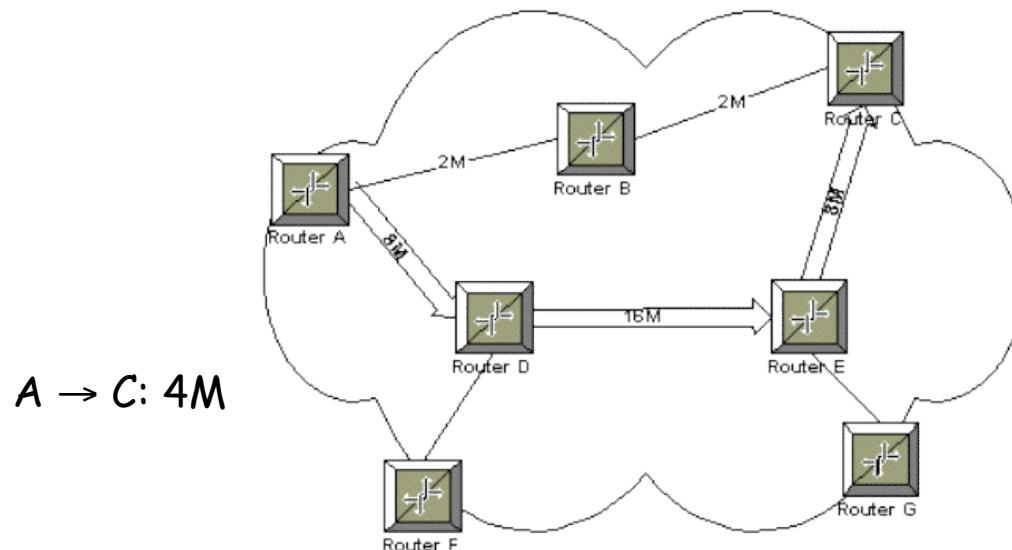


Scheduling

- Recursos compartidos
 - Buffer space
 - Capacidad en el enlace de salida
 - Tiempo de procesador
- Tipos de schedulers
 - Work-conserving
 - Non-work-conserving (no veremos)
- Schedulers sin prioridades
 - FCFS, RR, ...
- Schedulers con prioridades
 - GPS, WFQ, SCFQ, WF2Q, ...
- Características deseables
 - Sencillo de implementar
 - Reparto justo (*max-min fair share*) y protección
 - Performance bounds (deterministas o estadísticos)
 - Que permita implementar un CAC simple

QoS Routing

- Encontrar caminos “buenos” para flujos con requisitos específicos de QoS
- Usar la red de forma eficiente: aumentar la probabilidad de aceptar peticiones futuras
- Es complicado:
 - Información precisa sobre el estado de la red es difícil de mantener
 - Calcular caminos que cumplan requisitos de QoS es costoso (computacionalmente hablando)
- *Constraint-based Routing*
 - Calcular caminos teniendo en cuenta no solo QoS sino también políticas



Video en el acceso

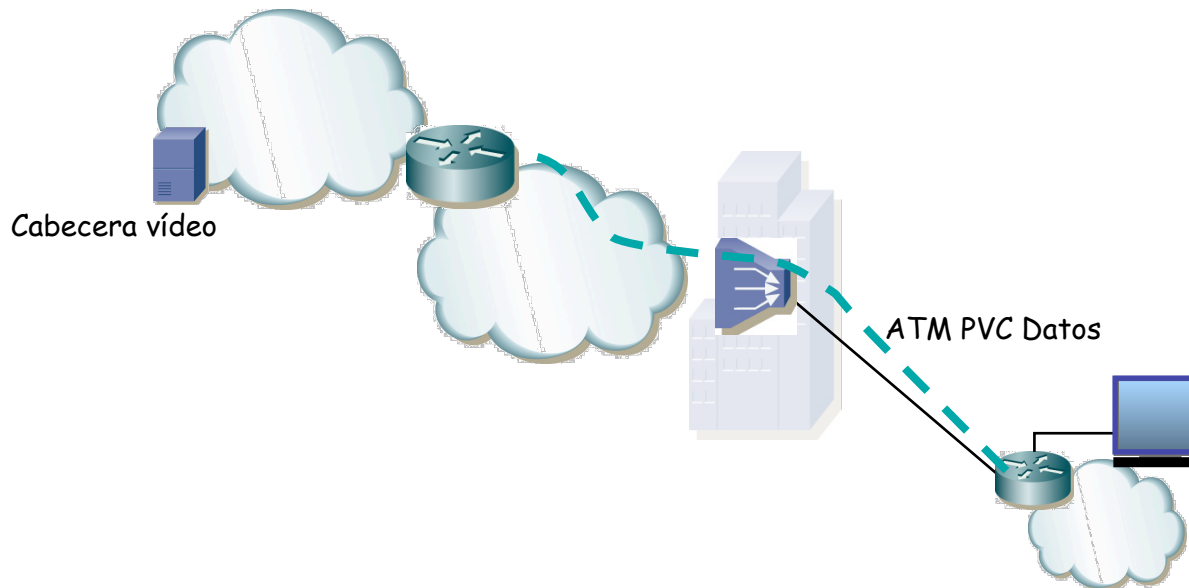
- Distribución MPEG transport stream
 - HFC
 - ADSL
 - FTTH
- Sobre canal de datos IP

Vídeo en la red de transporte

- Distribución MPEG transport stream
 - MPLS
 - ATM
- Transporte para el vídeo por canales independientes
 - GbE
 - ATM OC-192
 - WDM
 - ROADM

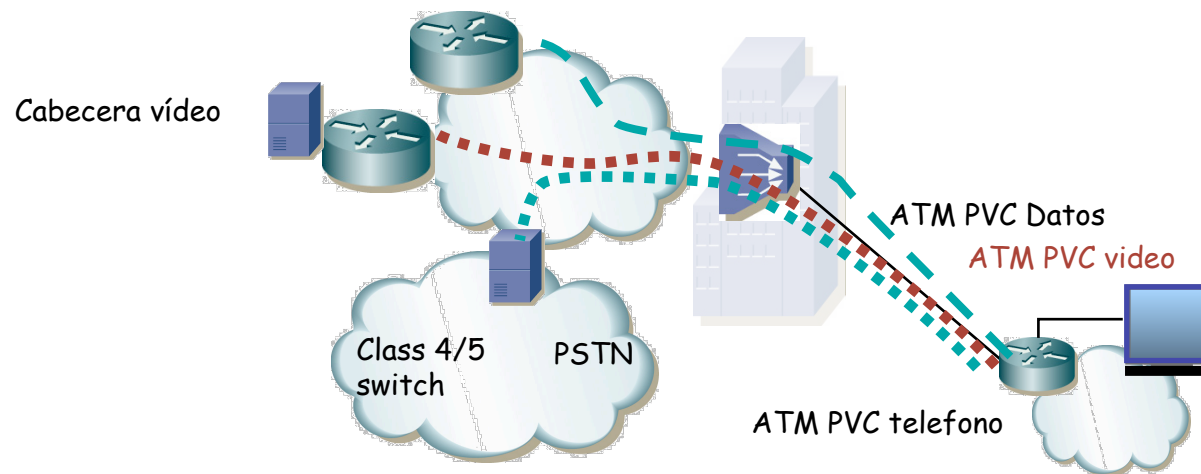
Ejemplo: acceso ADSL

- Un solo PVC para datos
 - Transporte de 1 canal de vídeo sobre ATM
 - Transporte de 1 canal de vídeo sobre red IP del operador



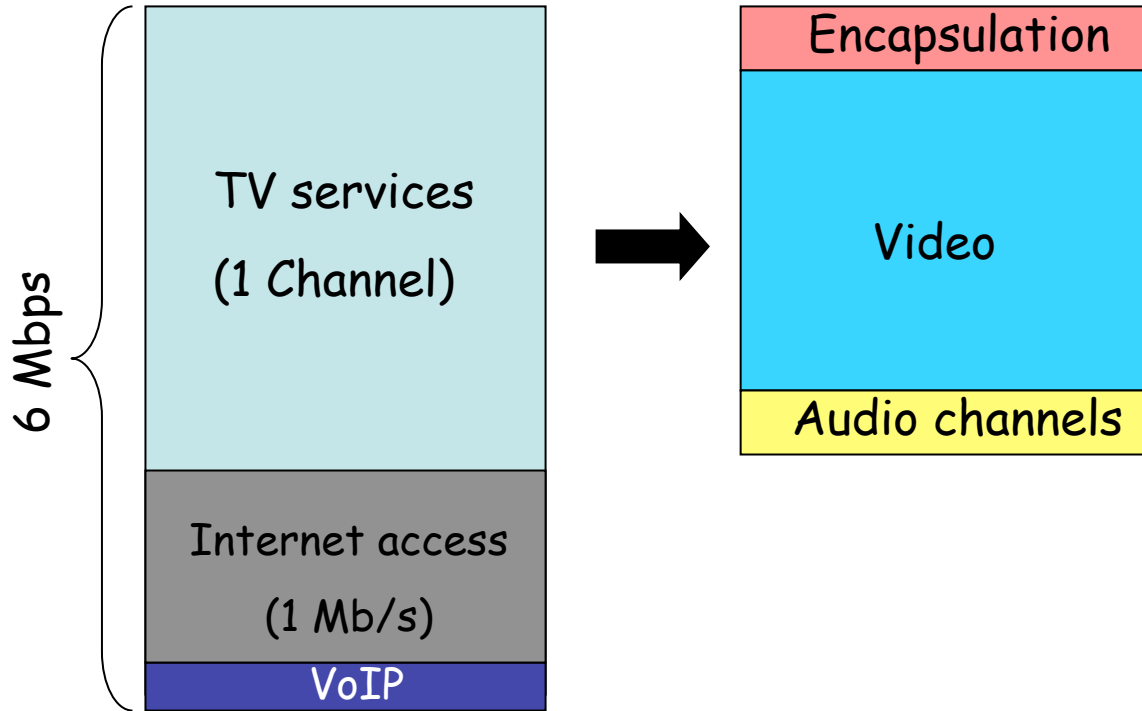
Ejemplo: acceso ADSL

- Acceso ADSL +1 VC para el vídeo
- Transporte de 1 canal de vídeo sobre ATM
 - Se puede enviar el canal a más usuarios con multipunto ATM



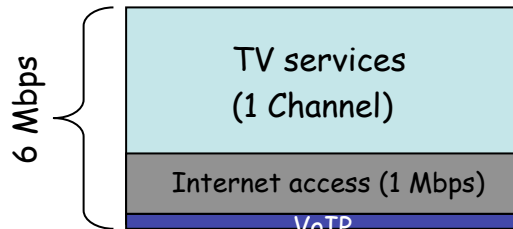
Red de acceso ADSL

Hoy

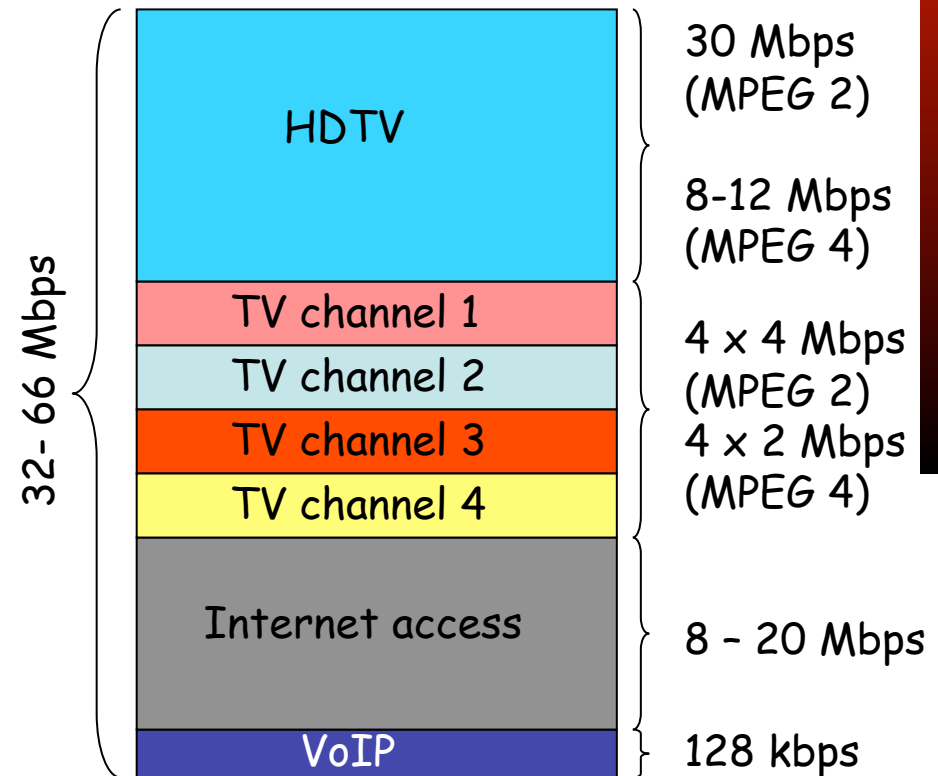


Red de acceso ADSL2+

Hoy



Futuro



Streaming

- Introducción -

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Programa de Tecnologías para la gestión distribuida
de la información