



1

## La capa de aplicación

Flujos de vídeo y redes de distribución de contenidos.

## Tema 2: La capa de aplicación

2

- 2.1 Principios de las aplicaciones de red.
- 2.2 La Web y HTTP (2ª parte).
- 2.3 Correo electrónico en Internet.
- 2.4 DNS: el servicio de directorio de Internet.
- 2.5 Distribución de archivos P2P.
- 2.6 Flujos de vídeo y redes de distribución de contenidos.**
- 2.7 Programación de sockets: creación de aplicaciones de red.

## Flujos de vídeo.

3

- Vídeo en Internet.
  - ▣ Representa la mayor parte del tráfico en los ISP residenciales.
  - ▣ Aplicaciones de flujos de vídeo almacenado son, p.e:
    - Vídeos almacenados en servidores y solicitados por usuarios (aplicaciones).
    - Películas, eventos deportivos, programas TV, vídeos generados por usuarios, etc.
- Características del vídeo.
  - ▣ Secuencia de imágenes que se deben visualizar a velocidad constante.
  - ▣ Imágenes codificadas digitalmente con elevada redundancia.
  - ▣ Se “comprime” para reducir la tasa de bits resultante → Cierta degradación de la calidad de las imágenes.
    - Mayor grado de compresión → Menor tasa de transmisión → Menor calidad de las imágenes.
    - Menor grado de compresión → Mayor tasa de transmisión → Mayor calidad de las imágenes.

## Flujos de vídeo.

4

- Vídeo comprimido para Internet.
  - ▣ Entre 100 kbps (vídeo de baja calidad) y más de 3Mbps (películas de alta resolución).
  - ▣ Produce elevado volumen de tráfico y de almacenamiento.
  - ▣ Como medida del rendimiento: tasa de transferencia media extremo a extremo (TTMEE).
  - ▣ Para reproducción continua:
    - La red debe proporcionar a la aplicación una TTMEE  $\geq$  tasa del vídeo comprimido.
  - ▣ Permite diferentes versiones del mismo vídeo con diferentes grados de calidad, p.e:
    - Tasas de 300 kbps, 1 Mbps y 3 Mbps.
    - Usuario puede seleccionar la versión a visualizar según su velocidad de conexión a Internet.

## Flujos de vídeo.

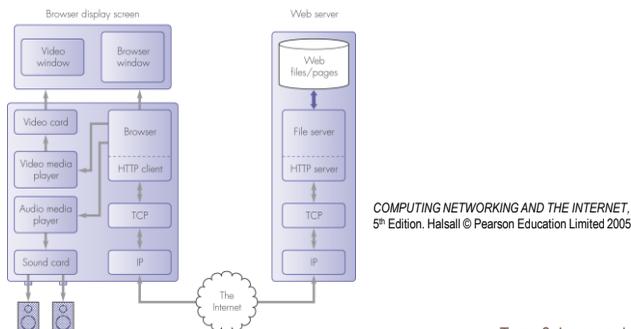
5

- Acceso a contenidos de vídeo vía Web.\*
  - ▣ Para pequeños ficheros:
    - Servidor Web convencional, se descarga el fichero y luego se reproduce.
  - ▣ Para aplicaciones de “flujo continuo” en principio mediante:
    - Servidor Web convencional y metaarchivo.
    - Servidor de “flujo continuo”.
  - ▣ Ejemplos de aplicaciones de “flujo continuo”:
    - Transmisión de audio y vídeo diferido (almacenado): vídeo (y audio) bajo demanda.
    - Transmisión de audio y vídeo “en vivo” (no almacenado): difusión de radio y TV.

## Flujos de vídeo.

6

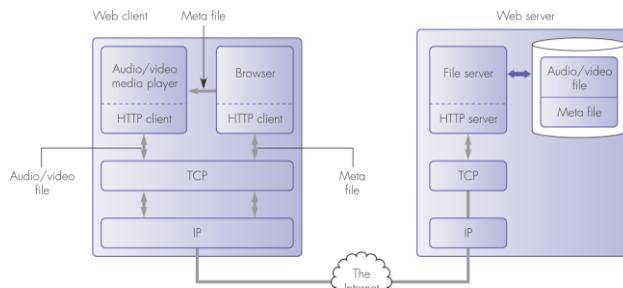
- Vídeo con servidor Web convencional.
  - ▣ Se procede de manera similar a cuando se descarga un fichero de texto/imagen.
  - ▣ Se descarga el fichero, se invoca la aplicación reproductora y se reproduce el contenido.
  - ▣ Inconvenientes.
    - Navegador primero descarga el fichero completo y luego lo pasa al reproductor.
    - Retardo elevado, admisible solo para ficheros muy pequeños.
    - No permite streaming (procesar/visualizar contenido mientras se recibe).



## Flujos de vídeo.

7

- Vídeo con servidor Web convencional y metafichero.
  - Requiere metafichero (MF) asociado a cada fichero de vídeo en el servidor Web.
  - MF contiene URL del fichero de *medios* y la especificación del tipo de contenido.
  - Hiperenlace en la página Web contiene el URL del MF, no el URL del vídeo.
  - Fichero de vídeo se descarga directamente al reproductor, no interviene el navegador.
  - Permite *streaming* (procesar contenido mientras se recibe).



El reproductor interactúa con el servidor, pero con HTTP...

COMPUTING NETWORKING AND THE INTERNET, 5th Edition. Halsall © Pearson Education Limited 2005

## Flujos de vídeo.

8

- Procedimiento de descarga con MF.
  - Al "pinchar" el hiperenlace se desarrolla una transacción HTTP/TCP y se descarga el MF al navegador.
  - Navegador: utiliza valor de la cabecera "Content-Type" del mensaje HTTP de respuesta para invocar al reproductor y le pasa el MF.
  - Reproductor: a partir del URL del vídeo en el MF lo descarga mediante HTTP/TCP.
    - Aplica un retardo predefinido de varios segundos para que su *buffer* se llene parcialmente.
    - Descomprime y envía el flujo de vídeo al *hardware* para su reproducción.
- Ventaja.
  - Elimina el retardo del navegador cuando éste accede directamente al fichero de vídeo.
- Inconvenientes.
  - Elevado retardo, se accede al vídeo igual que a un fichero de texto/imagen (HTTP/TCP)
    - TCP transfiere en segmentos y desarrolla control de errores, de congestión y de flujo.
    - Se requiere de *buffers* más grandes para "ocultar" la pérdida de un segmento.
  - Interacción NO satisfactoria entre usuario y servidor.

## Flujos de vídeo

9

- Modo “flujo continuo”.
  - ▣ Modo adecuado para la descarga de vídeos, excepto para secuencias muy cortas.
  - ▣ Requiere servidores de “flujo continuo”.
  
- Servidores de “flujo continuo”.
  - ▣ Optimizados para altas tasas de transferencia de flujos continuos de vídeo y acceso simultáneo de muchos usuarios.
  - ▣ Adecuados para aplicaciones de vídeo a través de PC, TV o equipo multimedia con conexión de alta velocidad a la red.
  - ▣ Normalmente evitan HTTP y/o TCP.
  - ▣ Permiten:
    - Modo *streaming* (procesar contenido mientras se recibe).
    - Buena interacción entre reproductor y servidor.

## Flujos de vídeo.

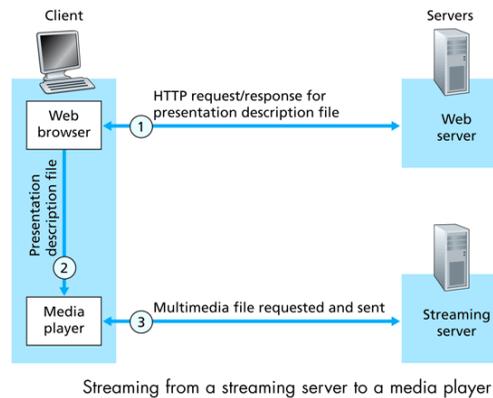
10

- Vídeo con servidor de flujo continuo.
  - ▣ Requiere dos servidores, que pueden o no compartir el mismo sistema físico.
    - Servidor Web, se ocupa de las páginas Web y de los metaarchivos.
    - Servidor de vídeo, se ocupa de los archivos de vídeo.
  
  - ▣ Protocolo de transporte: UDP.
    - Para evitar los retardos que introducen los procedimientos TCP.
    - Generalmente de manera conjunta con RTP (*Real-Time Transport Protocol*)
    - RTP/UDP para la transferencia de vídeo.
  
  - ▣ Protocolo de control: RTSP, *Real-Time Streaming Protocol*.
    - Protocolo para el control del proceso de reproducción.
    - RTSP/UDP para diálogo de control entre reproductor y servidor de vídeo.
  
  - ▣ Inconvenientes.
    - Requiere dos servidores (servidor Web y servidor RTSP).
    - Solución más costosa, sobre todo para implantaciones a gran escala a través de Internet.

## Flujos de vídeo.

11

### Vídeo con servidor de flujo continuo.



El reproductor interactúa con el servidor de vídeo con protocolos adecuados: RTP, RSTP, UDP.

## Flujos de vídeo.

12

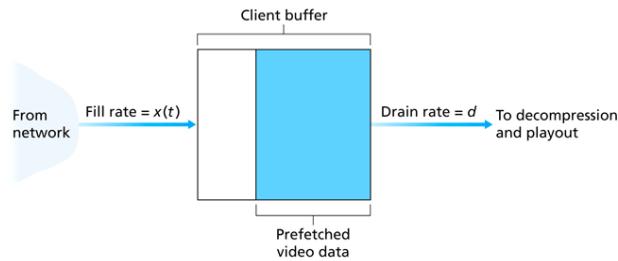
- Opciones para el envío de vídeo con servidor de flujo continuo.
  - Sobre UDP.
    - Receptor retrasa la reproducción unos pocos segundos para absorber el  *jitter* .
    - Requiere  *buffer*  asociado al reproductor del cliente.
    - Solución requerida en  contextos NO controlados  como  Internet .
  - Sobre UDP a tasa constante, ídem a la tasa de consumo del reproductor.
    - Receptor NO retrasa la reproducción.
    - No requiere  *buffer*  asociado al reproductor del cliente.
    - Solución posible en contextos controlados,  NO en Internet .
  - Sobre TCP para mejor calidad de la reproducción.
    - Receptor retrasa la reproducción unos pocos segundos para absorber el  *jitter* .
    - Requiere  *buffer*  asociado al reproductor del cliente.
      - El  *buffer*  se puede vaciar debido a los mecanismos de control de TCP → Pausas indeseables en la reproducción del vídeo → Necesidad de  buffer grande  para evitar que se vacíe.
    - Solución posible en contextos controlados,  NO en Internet ,  excepto :
      - Si se emplea la técnica de "precarga",  válida solo para vídeo almacenado . P.e. Youtube, Netflix.
      - Precarga : cliente descarga el vídeo, si puede, a una tasa superior a la tasa de reproducción (ver Kurose - Ross, 7ª edición, capítulo 9, pág. 568 final y pág. 569).

## Flujos de vídeo.

13

### Buffer de recepción en la aplicación reproductora de vídeo

Una vez cargado el *buffer* (durante unos pocos segundos) se inicia la extracción del vídeo seguida de la descompresión y la reproducción.



## Flujos de vídeo.

14

- Tecnología DASH, *Dynamic Adaptive Streaming over HTTP*.
  - Consecuencia de que los flujos de vídeo con HTTP presentan una carencia fundamental:
    - Todos los usuarios reciben la misma versión codificada del vídeo.
    - No todos los usuarios disponen del mismo ancho de banda (AB).
  - Solución: tecnología DASH.
    - El vídeo se codifica en diferentes versiones → Diferentes tasas, diferentes calidades.
    - Usuario solicita de manera dinámica con mensajes GET uno a uno segmentos de vídeo de pocos segundos de duración, según AB disponible para el usuario.
      - Cuando el AB es grande, solicita segmentos de vídeo de tasa elevada (elevada calidad).
      - Cuando el AB es pequeño, solicita segmentos de vídeo de tasa baja (baja calidad).
    - Permite adaptarse a las variaciones del AB durante una sesión (p.e, en entornos móviles).
    - Requiere archivo de "manifiesto" en el servidor HTTP:
      - Contiene para cada versión del vídeo el URL y la tasa de bits correspondiente.

## Flujos de vídeo.

15

- Procedimiento seguido por el cliente con tecnología DASH.
  - ▣ Solicita el archivo de “manifiesto”.
  - ▣ “Aprende” las diferentes versiones disponibles para el vídeo en cuestión.
  - ▣ Solicita segmentos de vídeo de uno en uno:
    - Mediante mensaje GET de HTTP para cada segmento.
    - Especificando URL y rango de bytes de cada segmento.
  - ▣ El cliente, durante la descarga del vídeo:
    - Mide el AB de recepción que está experimentando.
    - Determina qué tasa solicitar para la descarga del siguiente segmento.
    - Consecuencia: la calidad de la descarga varía en función de lo anterior.

## Redes de distribución de contenidos.

16

- CDN, *Content Distribution Network*.
  - ▣ Solución para distribución eficiente de contenidos a través de Internet (p.e. vídeos).
  - ▣ Solución para acercar de manera eficiente los contenidos a los usuarios.
  - ▣ Supera los problemas de una solución centralizada con “centro de datos”.
- Problemas de una solución centralizada.
  - ▣ Único punto de fallo.
    - Si “cae” el centro de datos y/o sus enlaces a Internet → Imposible distribuir contenidos.
  - ▣ Mayoría de clientes muy alejados del centro de datos.
    - Muchos ISP involucrados (incluso en diferentes continentes).
    - Mayor probabilidad de enlaces “cuello de botella”.
    - Mayor probabilidad de TTME < Tasa del vídeo → “Congelaciones” de la imagen.
  - ▣ Uso ineficiente de los recursos de transmisión.
    - Muchos enlaces enviando una y otra vez los mismos flujos de bytes para vídeos muy populares.

## Redes de distribución de contenidos.

17

- CDN, funciones.
  - ▣ Gestión de servidores situados en múltiples ubicaciones distribuidas geográficamente.
    - Grupos (*clusters*) de servidores que almacenan copias de los contenidos (vídeos, audios, imágenes, documentos, etc.).
    - Replicación de contenidos en numerosos servidores.
  - ▣ Redireccionado de cada petición al *cluster* de servidores que posiblemente proporcione la mejor experiencia al usuario.
  
  - ▣ Una CDN no siempre almacena en cada *cluster* una copia de cada contenido.
    - Hay contenidos que se solicitan poco y/o son populares sólo en ciertos países/regiones.
  - ▣ Estrategia para contenidos solicitados y no disponibles en un *cluster*:
    - El *cluster* solicita dicho contenido a un repositorio central o a otro *cluster*.
    - El *cluster* almacena una copia localmente.
      - Como en las cachés Web, si se llena el almacenamiento se eliminan los contenidos menos solicitados.
    - El *cluster* envía contenido al usuario.

## Redes de distribución de contenidos.

18

- CDN, soluciones.
  - ▣ Privadas, son propiedad de un proveedor de contenidos (p.e, CDN de Google).
  
  - ▣ Comerciales, distribuyen contenidos de múltiples proveedores (p.e, de Akamai).
  
  - ▣ En cuanto a la ubicación de los servidores:
    - Introducción profunda → Elevada distribución de servidores.
      - Objetivo: aproximarse a los usuarios finales para mejorar el retardo y la TTMEE.
      - Implantación de *clusters* de servidores en los ISP de acceso por todo el mundo.
      - Muy complejo mantener y gestionar tantos *clusters* de servidores.
    - Atraer a los ISP → Reducida distribución de servidores.
      - Grandes *clusters* de servidores en un reducido nº de ubicaciones.
        - P.e, en una decena de IPX (*Internet Exchange point*).
      - Menos complejo mantener y gestionar los *clusters*.
      - Servidores más alejados de los usuarios → Posiblemente a costa del retardo y la TTMEE.

## Redes de distribución de contenidos.

19

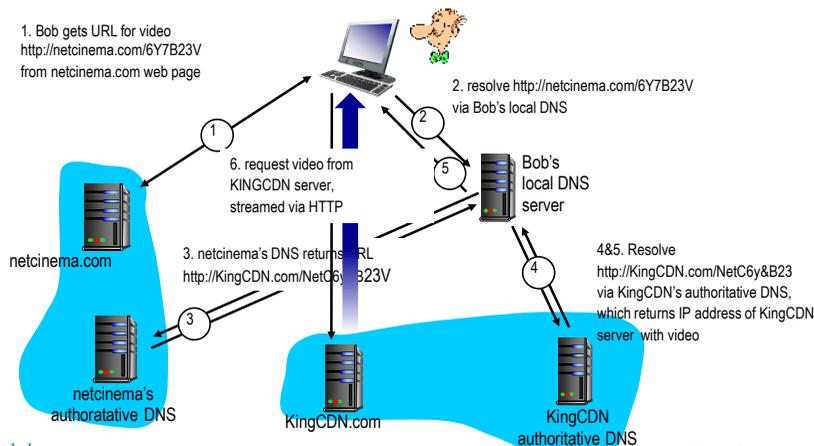
- CDN, funcionamiento.
  - ▣ Cuando se solicita un vídeo mediante su URL, la CDN debe interceptar la solicitud para:
    - Determinar el *cluster* de servidores más adecuado para el usuario en ese momento.
    - Redirigir la solicitud a un servidor ubicado en el *cluster* seleccionado.
  - ▣ La mayoría de CDN aprovechan el servicio DNS para interceptar y redirigir las solicitudes.
  - ▣ Ejemplo de intersección y re direccionamiento de una solicitud (fig. 2.25, pág. 126).

## Redes de distribución de contenidos.

20

- CDN, funcionamiento: ejemplo de intersección y re direccionamiento de una solicitud.
  - ▣ Cliente (Bod) solicita el vídeo <http://netcinema.com/6Y7B23V>.
  - ▣ Vídeo almacenado en <http://KingCDN.com/NetC6y&B23V>.

**NetCinema:** empresa proveedora de contenidos.  
**KingCDN:** empresa proveedora de servicios CDN.



## Redes de distribución de contenidos.

21

- CDN, estrategia de selección de *clusters*.
  - Mecanismo para dirigir de manera dinámica las solicitudes de contenidos hacia un *cluster* de servidores o a un centro de datos de una CDN.
  - Usualmente:
    - A partir de la dirección IP del DNS local del usuario y mediante estrategias propietarias.
  - Ejemplos de estrategias propietarias:
    - Asignación del *cluster* más cercano geográficamente (a menos km “a vuelo de pájaro”).
      - Inconvenientes: [ver libro de texto, último párrafo - página 126](#).
    - Asignación del *cluster* según condiciones actuales del tráfico en la red.
      - Inconvenientes: [ver libro de texto, primer párrafo - página 127](#).
- Casos de estudio ([ver libro de texto](#)).
  - Infraestructura de red de Google ([recuadro, página 125](#)).
  - Netflix, YouTube y Kankan ([apartado 2.6.4, páginas 127 – 130](#)).