

# Capítulo 7

## Multimedia en Redes de Computadores

*Material tomado de:  
Computer Networking: A Top Down Approach  
Featuring the Internet,  
Jim Kurose, Keith Ross.*

# Capítulo 7: Contenidos

- 7.1 Aplicaciones Multimedia en Red
- 7.2 Streaming de Audio y video almacenado
- 7.3 Real-time Multimedia: Estudio de telefonía en Internet
- 7.4 Distribución de Multimedia: Redes de distribución de Contenidos
- 7.5 protocolos para aplicaciones Interactivas de Tiempo Real
  - RTP, RTCP, SIP
- 7.6 Más allá de Best Effort
- 7.7 Mecanismos de itineración y políticas
- 7.8 Servicios Integrados y Servicios Diferenciados
- 7.9 RSVP

# Voice-over-IP (VoIP)

- ❑ *Requerimientos extremo-a-extremo en VoIP*: aspectos necesarios para mantener una conversación:
  - Altos retardos son notorios, dificulta interactividad
  - < 150 msec: bueno
  - > 400 msec malo, debe evitars asumiendo pérdida
  - Retardo debe incluir retardos de la aplicación (paquetizar, reproducción) y de la red
- ❑ *Iniciación de la sesión*: cómo se da aviso de la IP, puerto, algoritmo de codificación?
- ❑ *Servicios de valor agregado*: reenvío de llamados, grabación
- ❑ *Servicios de emergencia*: 911 (USA) 131, 132, 133 (Chile)

# Multimedia Interactiva: Teléfono Internet

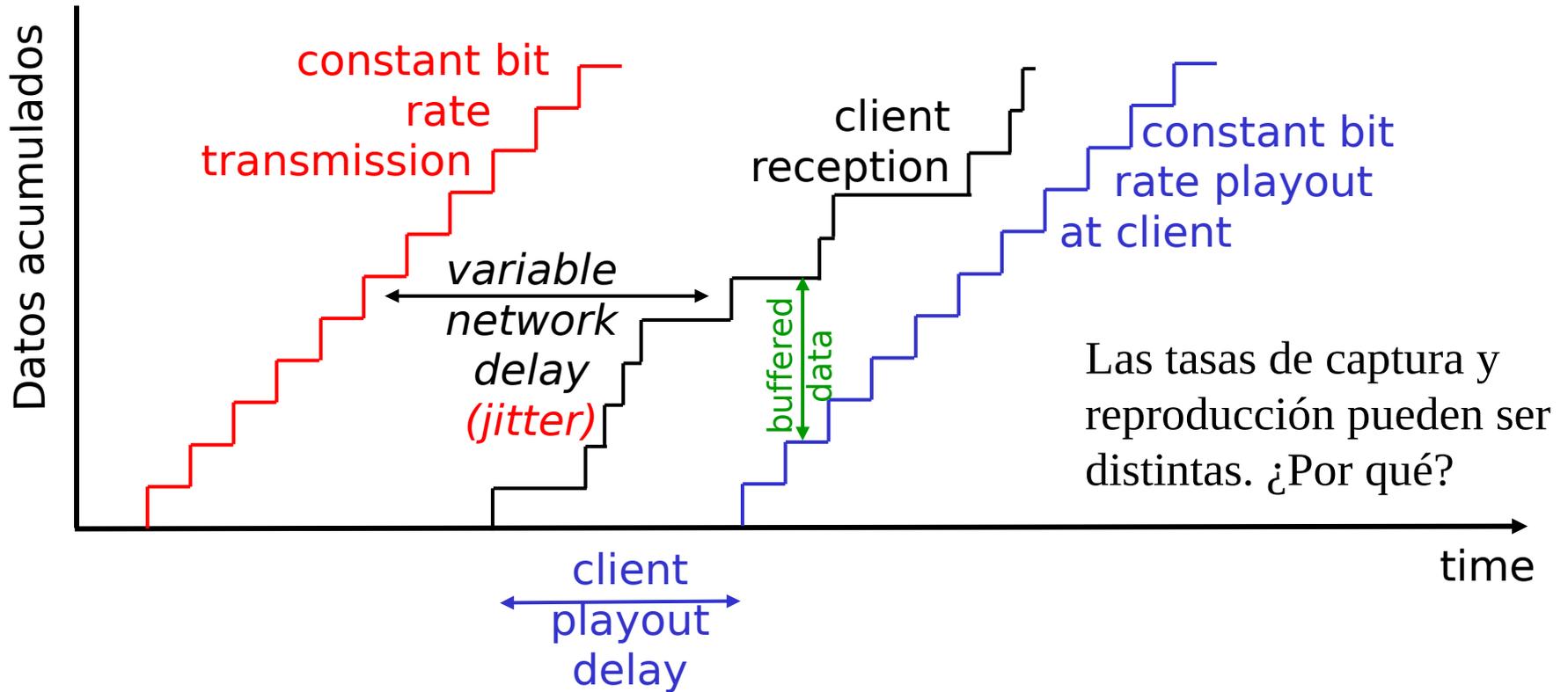
## Introduciremos Teléfono Internet a través de un ejemplo

- ❑ Audio emisor: alterna habla con periodos de silencio.
  - 64 kbps durante momentos de habla
  - Paquetes son generados sólo durante el habla
  - Segmento de 20 msec a 8 Kbytes/sec: 160 bytes de datos
- ❑ Encabezado capa aplicación es agregado a cada segmento (Protocolo RTP).
- ❑ Segmento + encabezado es encapsulado en datagrama UDP.
- ❑ Aplicación envía datagrama UDP por el socket cada 20 ms durante habla.

# Teléfono Internet: Pérdidas y retardo

- ❑ **Pérdidas en la red:** pérdida de datagrama IP debido a congestión en la red (overflow de buffer de router)
- ❑ **Pérdida por retardo:** Datagrama IP llega muy tarde para su reproducción en el receptor
  - retardo: procesamiento, colas en red; retardo en sistemas extremos (Tx y Rx)
  - Retardo máximo tolerable típico: 400 ms
- ❑ **Tolerancia a pérdidas:** dependiendo de codificación de voz, se puede tolerar entre 1% y 10% de paquetes perdidos.

# Variaciones del retardo (Delay Jitter)



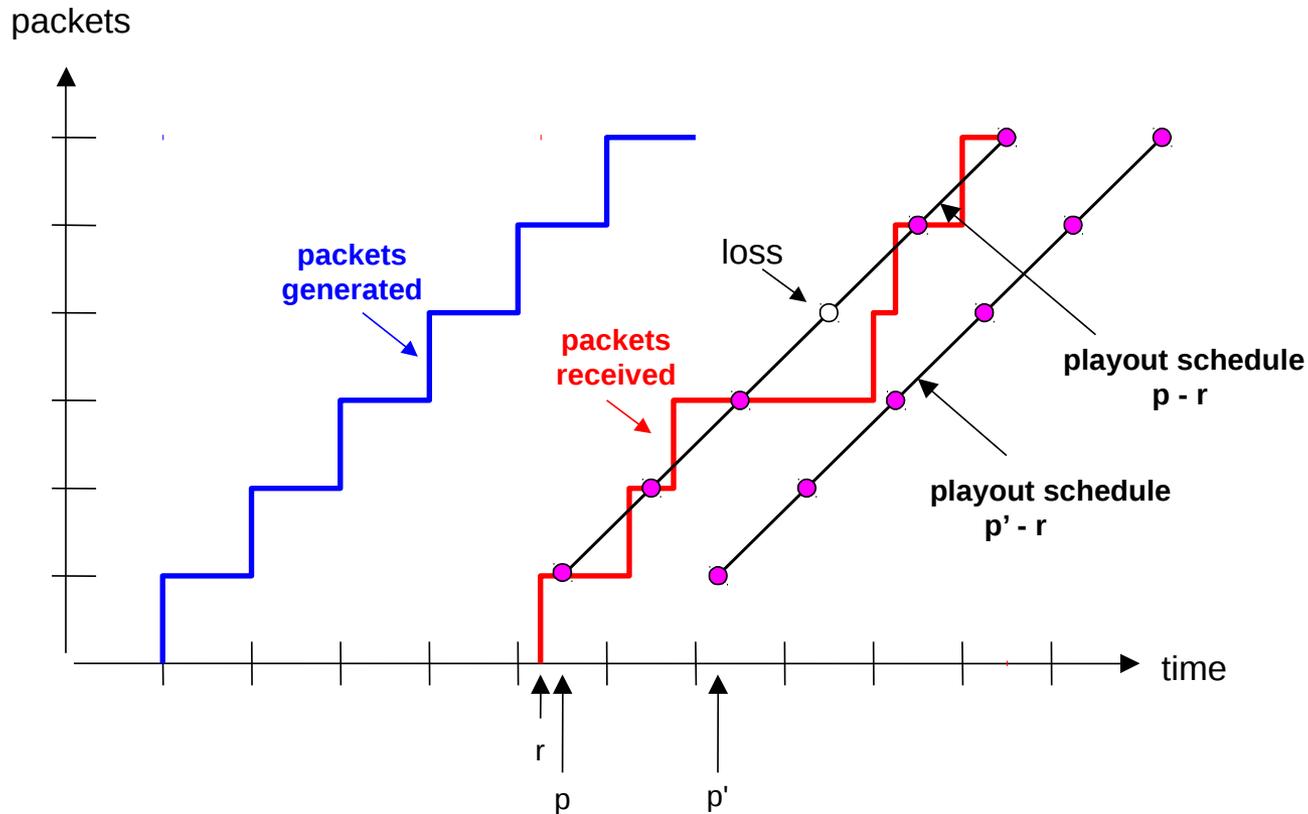
- Consideremos retardo extremo a extremo de dos paquetes consecutivos: **diferencia** puede ser más o menos de 20 ms

# Teléfono Internet: Retardo de reproducción fijo

- Receptor intenta reproducir cada golpe de habla exactamente  $q$  [ms] después que el habla fue generada.
  - habla tiene marca de tiempo  $t$ : reproducir después de  $t+q$  .
  - Habla llega después de  $t+q$ : datos llegan muy tarde para reproducción, datos son “perdidos”
- Compromiso para  $q$ :
  - $q$  grande: menor pérdida de paquete, más retardo
  - $q$  pequeño: mejor experiencia interactiva

# Retardo de reproducción fijo

- Tx genera paquetes cada 20 ms durante habla.
- Primer paquete recibido en tiempo  $r$
- Primer itinerario de reproducción: comienza en  $p$
- Segundo itinerario de reproducción: comienza en  $p'$



# Retardo de reproducción Adaptativo, I

- **Objetivo:** minimizar retardo de reproducción, manteniendo baja la tasa de pérdida por retardo
- **Estrategia:** Ajuste del retardo de reproducción adaptativo:
  - Retardo de red estimado, ajustar el retardo de reproducción al comienzo de cada segmento de habla.
  - Periodos de silencio alargados o comprimidos.
  - Habla aún reproducida cada 20 ms durante su presencia.

$t_i$  = marca de tiempo de  $i^{\text{mo}}$  paquete

$r_i$  = tiempo recepción paquete  $i$

$p_i$  = tiempo paquete  $i$  es reproducido (*playout*)

$r_i - t_i$  = retardo de red para  $i^{\text{mo}}$  paquete

$d_i$  = retardo promedio estimado después de recibir  $i^{\text{mo}}$  paquete

Estimación dinámica de retardo promedio en receptor, ojo con valores que adopta:

$$d_i = (1 - u) d_{i-1} + u (r_i - t_i)$$

Donde  $u$  es una constante fija (e.g.,  $u = .01$ ).

# Retardo de Reproducción Adaptativo II

- También es útil estimar el promedio de las variaciones de retardo,  $v_i$ :

$$v_i = (1 - u)v_{i-1} + u |r_i - t_i - d_i|$$

- Los estimadores  $d_i$  y  $v_i$  son calculados para cada paquete recibido, aún cuando ellos son usados sólo al inicio de cada segmento de habla.

- El primer paquete de un segmento de habla es reproducido en tiempo:

$$p_i = t_i + d_i + Kv_i$$

Donde  $K$  es una constante positiva (ej. 4).

- Si la reproducción es bajo demanda o en vivo no interactiva, podemos usar mayor  **$K$**
- Paquetes restantes dentro de un segmento de habla son reproducidos periódicamente (por ejemplo, cada 20 [ms]).

## Reproducción adaptativo, III

**Q:** Cómo el receptor determina que un paquete es el primero en un segmento de habla?

- Si no hay pérdida, receptor mira marcas de tiempo sucesivas.
  - Diferencia de marcas de tiempo sucesivas  $> 20$  ms  $\Rightarrow$  comienza nuevo segmento de habla.
- Con posible pérdida, receptor debe mirar las marcas de tiempo y números de secuencia.
  - Diferencia de marcas de tiempo sucesivas  $> 20$  ms **y** números de secuencia sucesivos espacios  $\Rightarrow$  segmento de habla comienza. Ésta es la regla general.
- Se requiere detección del habla en transmisor. Más adelante.

# Recuperación de pérdidas de paquetes (1)

## forward error correction (FEC): esquema simple

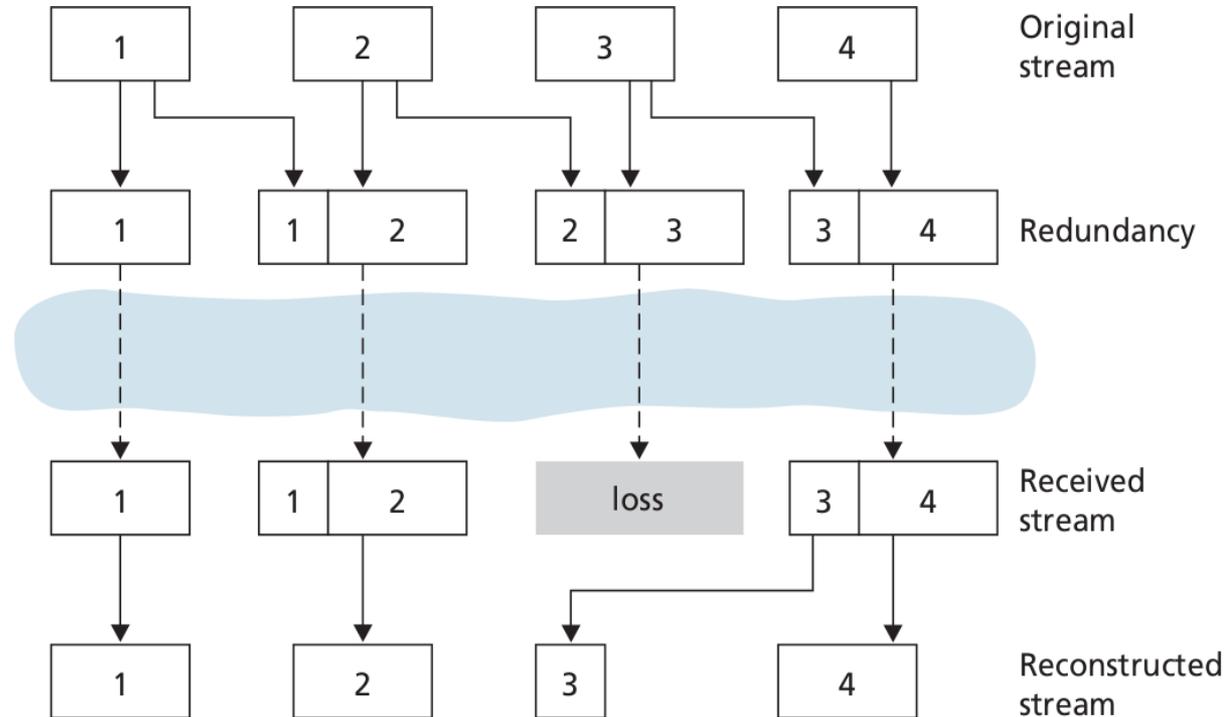
- ❑ Por cada  $n$  paquetes crea un paquete redundante dando paridad
- ❑ envía  $n+1$  paquetes, aumenta ancho de banda ocupado en factor  $1/n$ .
- ❑ Se puede reconstruir los  $n$  paquetes originales si hay a lo más un paquete perdido de los  $n+1$
- ❑ Retardo de reproducción debe ser suficiente para recibir todos los  $n+1$  paquetes
- ❑ Hay compromiso:
  - aumentar  $n \Rightarrow$  menos BW perdido
  - aumentar  $n \Rightarrow$  mayor retardo de reproducción
  - aumentar  $n \Rightarrow$  mayor probabilidad que 2 ó más paquetes se pierdan

# Recuperación de paquetes perdidos

## (2)

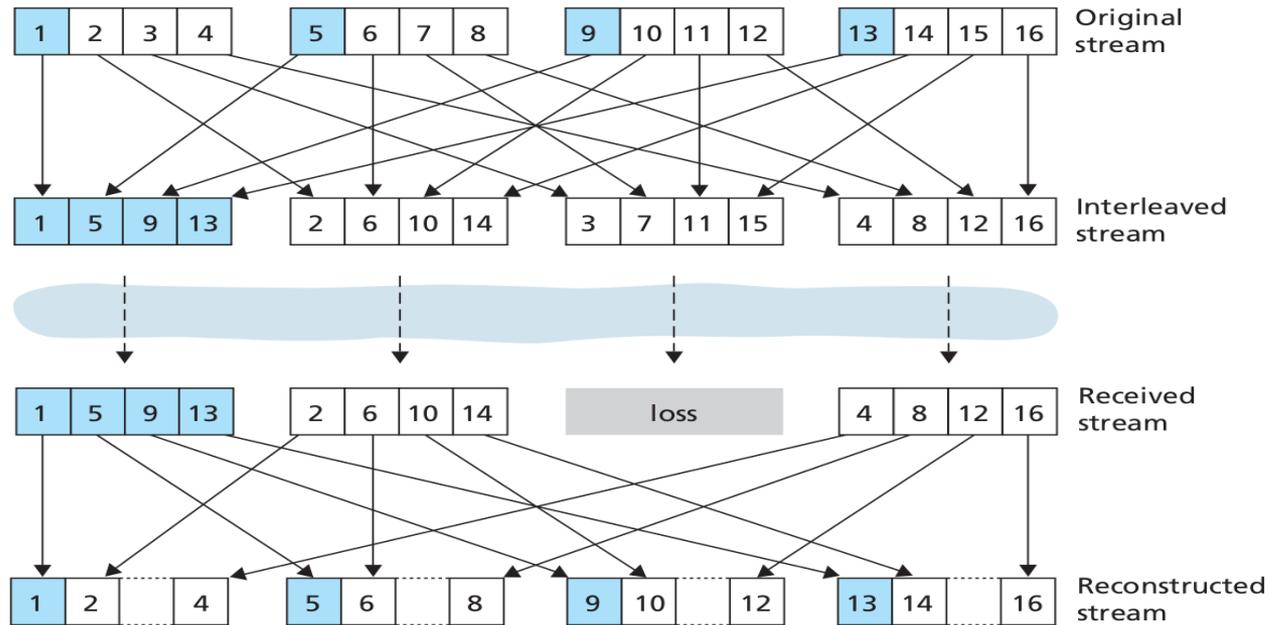
### 2º esquema FEC

- agrega un flujo de baja calidad
- envía flujo de baja resolución como información redundante
- por ejemplo, flujo nominal PCM a 64 kbps y flujo redundante GSM a 13 kbps.



- Cuando no hay pérdidas consecutivas, el receptor puede subsanar la pérdida.
- Se puede agregar también las tramas de baja calidad (n-1) y (n-2)

# Recuperación de pérdida de paquetes (3)



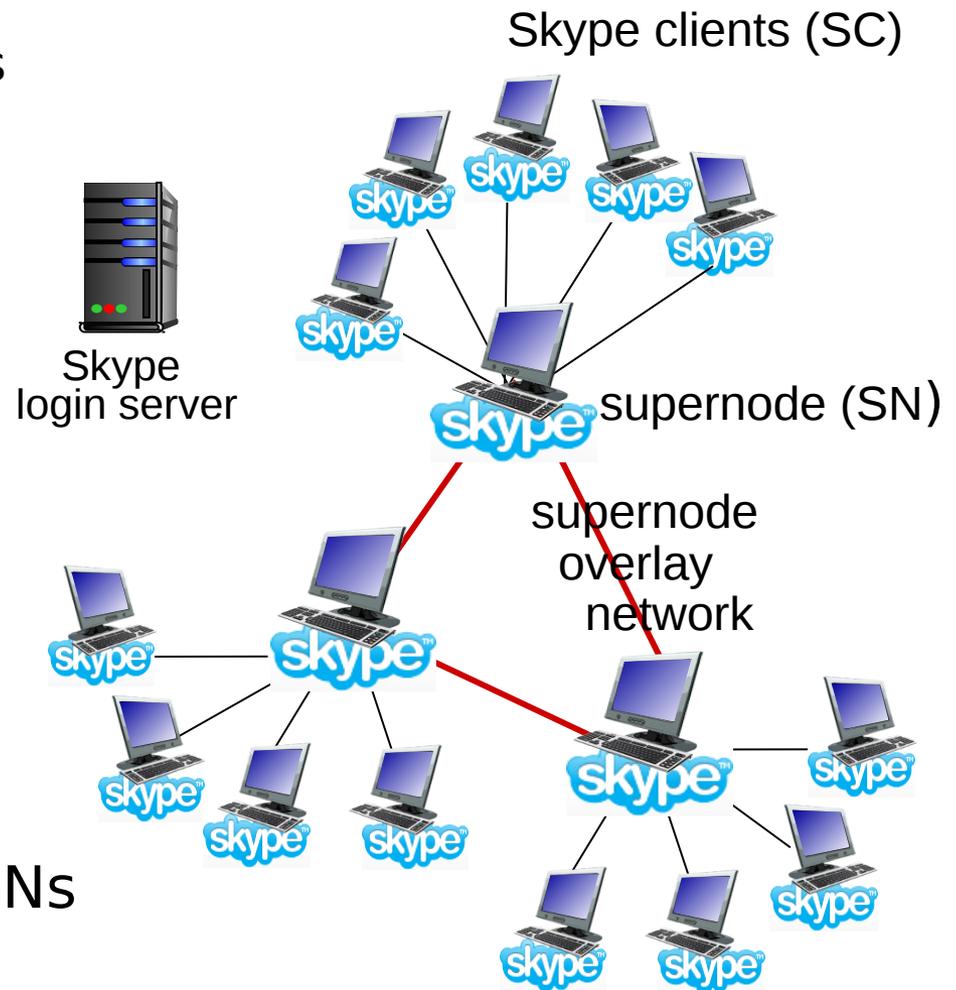
## Entrelazado

- ❑ Tramas son subdivididas en pequeñas unidades
- ❑ Por ejemplo, unidades de 4 ó 5 ms
- ❑ Paquete contiene pequeñas unidades de tramas diferentes

- ❑ Si paquete se pierde, aún se tiene la mayoría de cada trama
- ❑ No hay redundancia
- ❑ Se agrega retardo de reproducción

# Caso estudio: Voice-over-IP: Skype

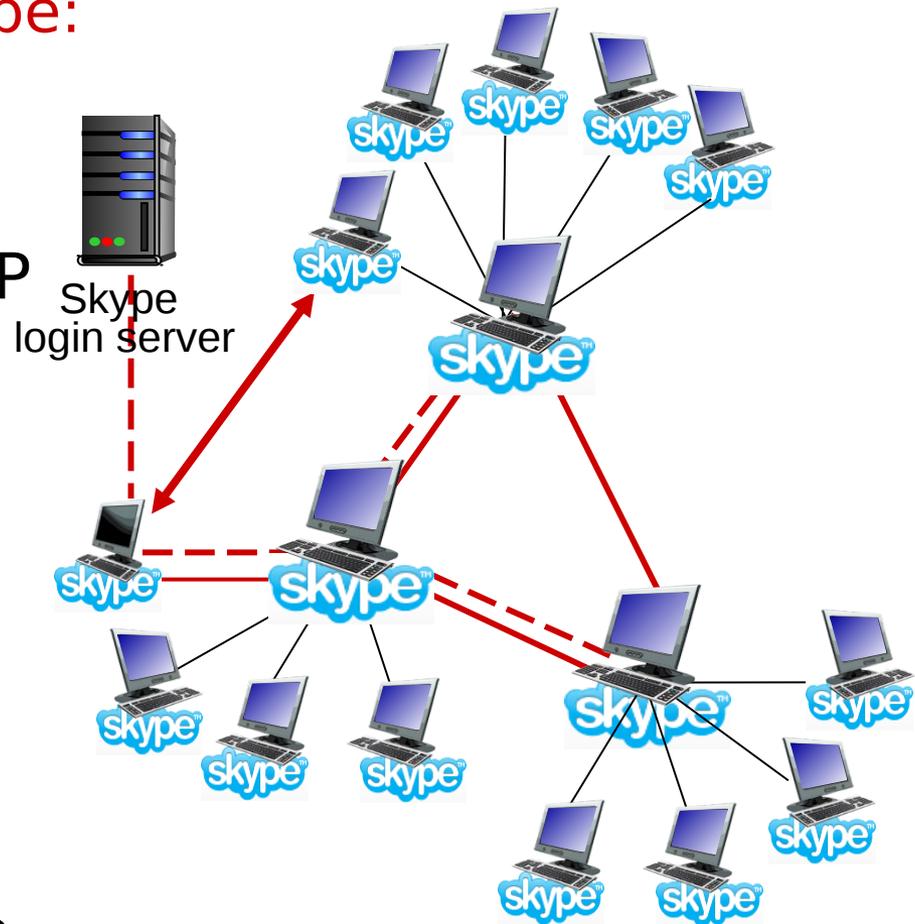
- ❖ Comprado por Microsoft el 2011 por sobre US\$8 billones
- ❖ Protocolo de aplicación propietario (inferido vía ingeniería inversa)
  - Mensajes encriptados
- ❖ Componentes P2P:
  - **Cientes (SC):** son los pares skype comunicados vía llamado VoIP
  - **super nodes (SN):** pares skype con funciones especiales
  - **overlay network:** entre SNs para localizar SCs
  - **login server**



# P2P voice-over-IP: skype

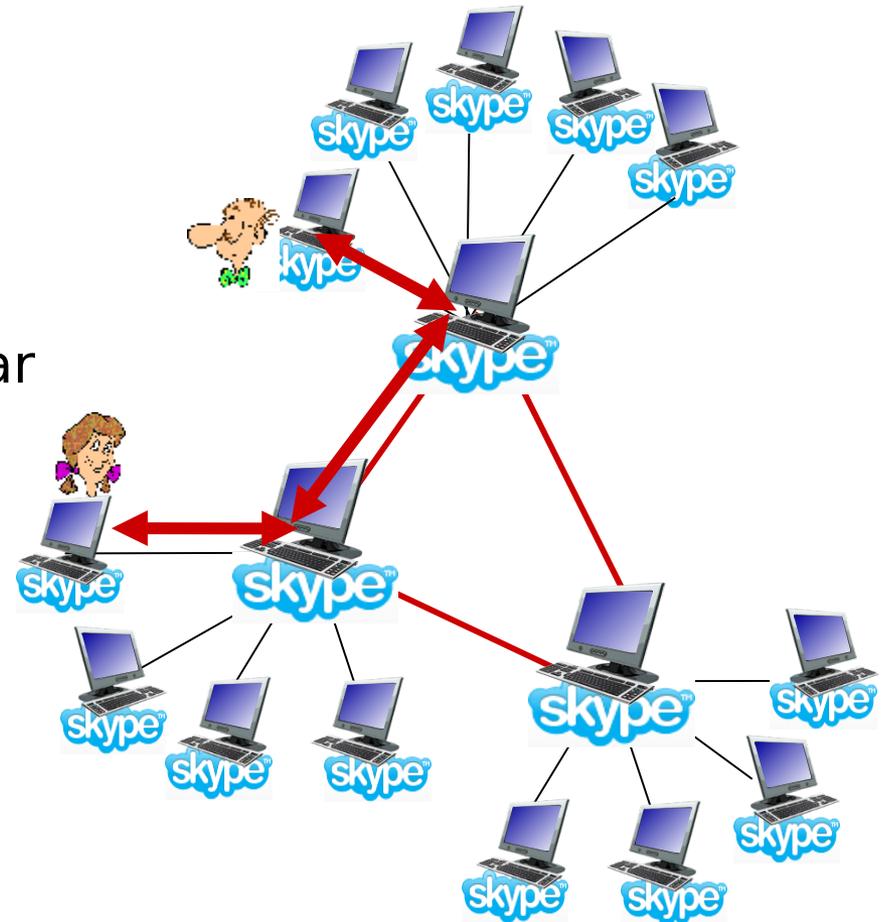
## Operación de clientes skype:

1. Ingresa a red contactando un SN (IP address cached) usa TCP
2. logs-in (username, password) a servidor central skype
3. obtiene IP de llamado desde SN, SN overlay
4. Inicia llamada directamente al llamado



# Skype: pares y relays

- **problema:** Alice y Bob están detrás de “NATs”
  - NAT previene inicio de conexiones a pares internos
  - Par interno sí puede iniciar conexión hacia afuera.
- ❖ **Solución relay:** Alice y Bob mantienen conexiones abiertas a sus SNs
  - Alice pide a su SN conectarse a Bob
  - SN de Alice se conecta a SN de Bob
  - SN de Bob se conecta a Bob por su conexión previamente abierta.



# Capítulo 7: Contenidos

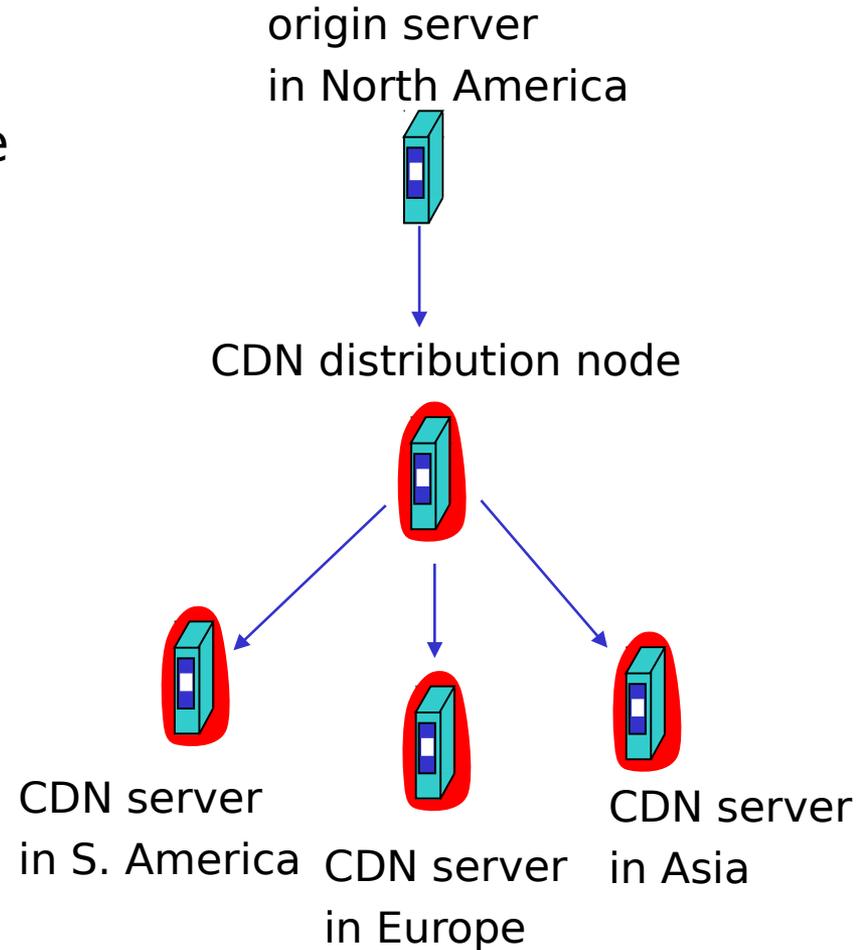
- ❑ 7.1 Aplicaciones Multimedia en Red
- ❑ 7.2 Streaming de Audio y video almacenado
- ❑ 7.3 Real-time Multimedia: Estudio de telefonía en Internet
- ❑ 7.4 Distribución de Multimedia: Redes de distribución de Contenidos
- ❑ 7.5 protocolos para aplicaciones Interactivas de Tiempo Real
  - RTP, RTCP, SIP
- ❑ 7.6 Más allá de Best Effort
- ❑ 7.7 Mecanismos de itineración y políticas
- ❑ 7.8 Servicios Integrados y Servicios Diferenciados
- ❑ 7.9 RSVP

# Redes de distribución de contenidos

## Content distribution networks (CDNs)

### Replicación de contenido

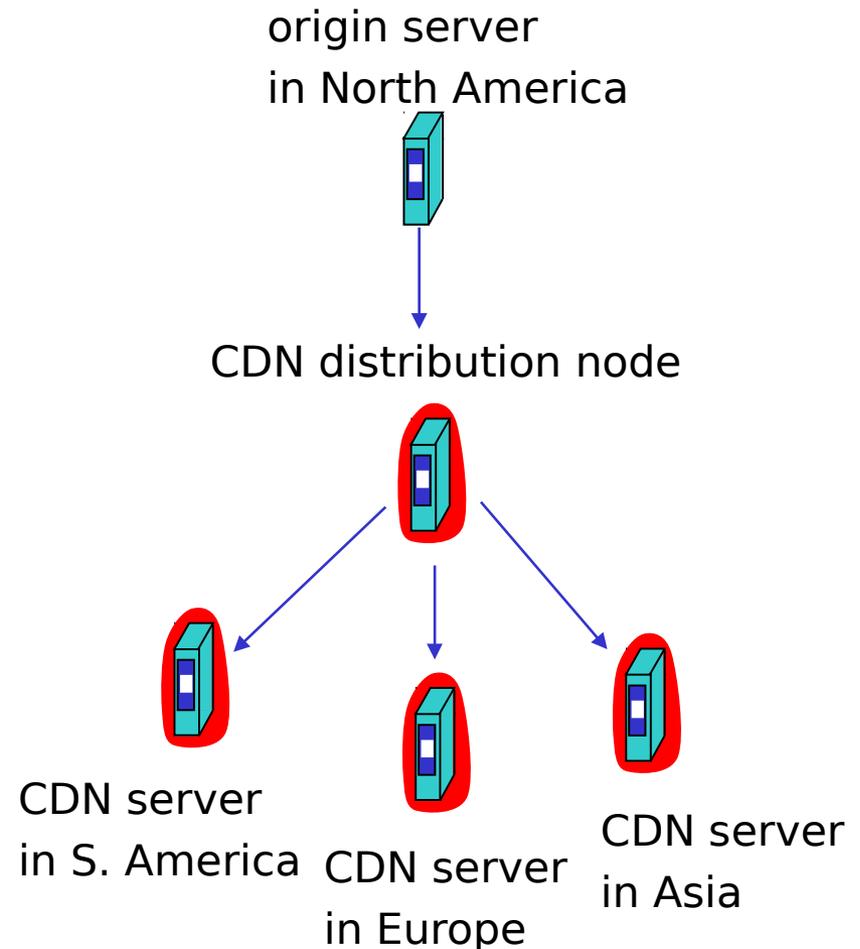
- ❑ Desafío: ¿Cómo envío gran archivo (e.g., video) desde único servidor de origen en tiempo real?
- ❑ Solución: replicar contenido en cientos de servidores a través de Internet
  - Contenido es bajado a servidor CDN con anticipación
  - Poner contenido “cerca” del usuario para evitar problemas (pérdidas, retardo) al enviar contenido sobre caminos más largos
  - Servidores CDN están típicamente en borde o red de acceso



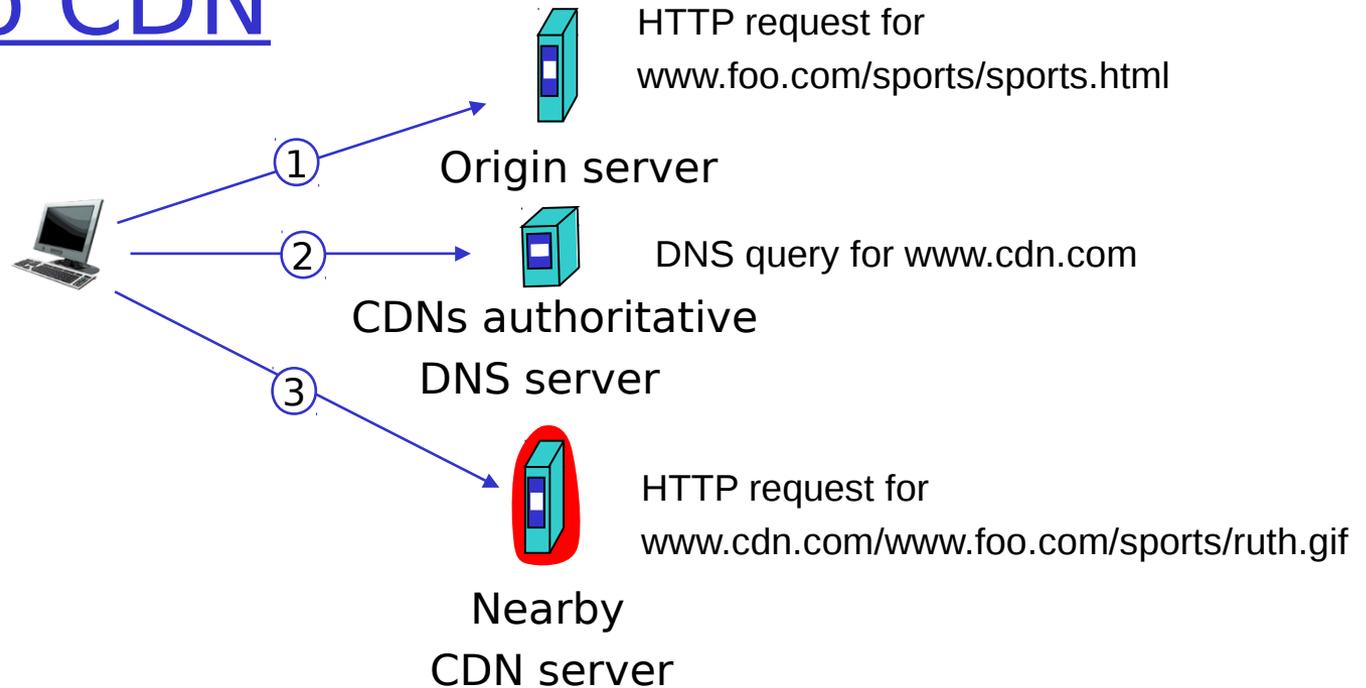
# Redes de distribución de contenidos (CDNs)

## Replicación de contenidos

- ❑ En CDN (e.g., Akamai) el usuario es el proveedor de contenidos (e.g., CNN)
- ❑ CDN replica el contenido del usuario en servidores CDN. Cuando el proveedor actualiza el contenido, CDN actualiza servidores



# Ejemplo CDN



## Servidor origen (www.foo.com)

- distribuye HTML

- reemplaza:

http://www.foo.com/sports/ruth.mpg

por

http://www.cdn.com/www.foo.com/sports/ruth.mpg

## Compañía CDN (cdn.com)

- Distribuye archivos mpg
- Usa su servidor DNS autoritario para redirigir los requerimientos

# Más sobre CDNs

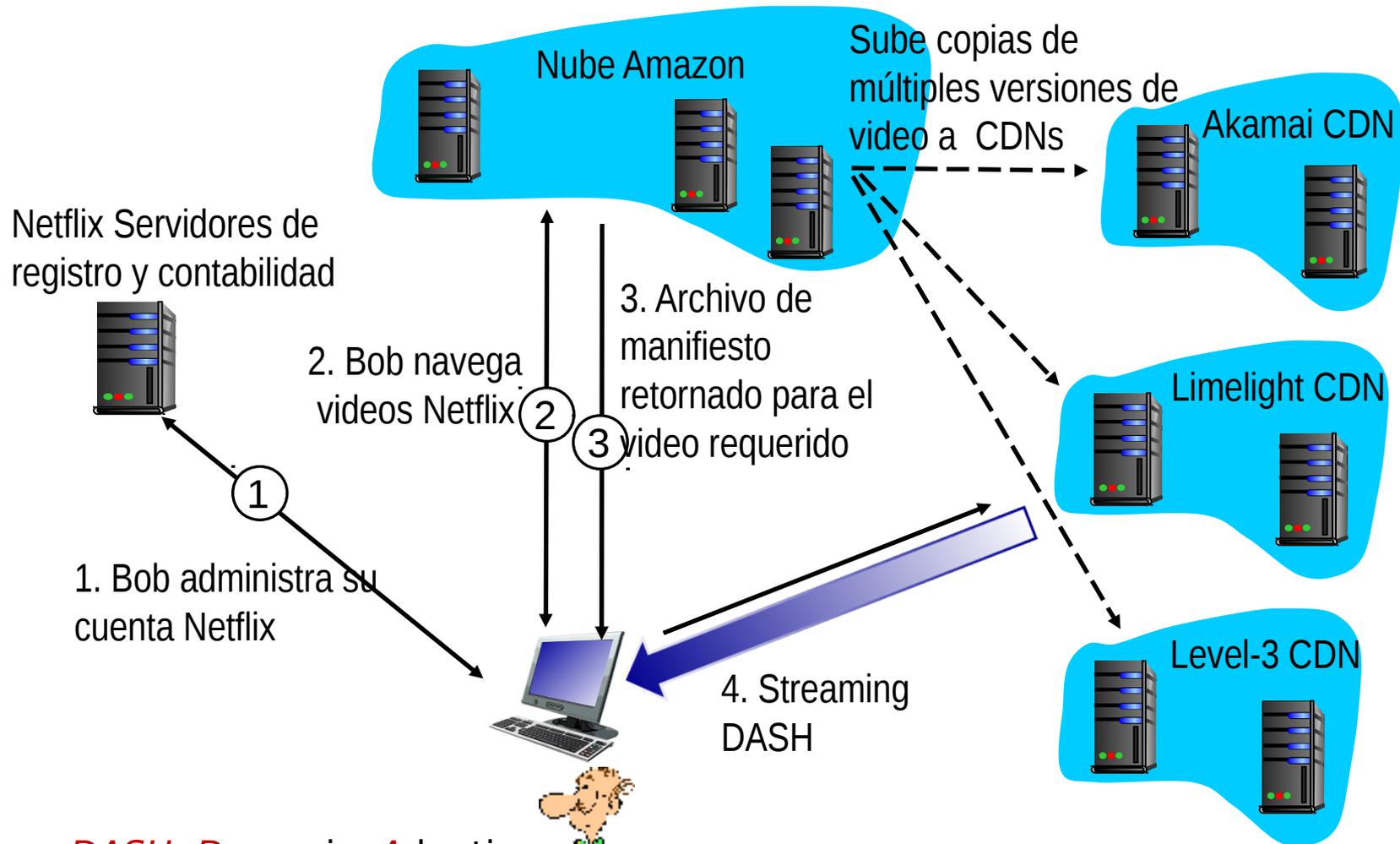
## Ruteo de requerimientos

- ❑ CDN crea un “mapa”, indicando distancias desde ISPs hojas y nodos CDN
- ❑ Cuando consulta llega a servidor DNS autoritario:
  - Servidor determina ISP desde el cual se origina la consulta
  - usa “mapa” para determinar mejor servidor CDN
- ❑ Nodos CDN crean red sobrepuesta en capa aplicación
- ❑ Alternativa: permitir que clientes decidan el servidor a partir de una lista de servidores CDN.
  - Cliente elige el mejor (usando ping)
  - Ésta es estrategia de Netflix

# Caso de estudio: Netflix

- Equivale al 30% del tráfico downstream en USA el 2011
- Dueña de muy poca infraestructura, usa servicios de terceros:
  - Dueña de registro y servidor de pago
  - Usa nube de Amazon (3<sup>rd</sup> party):
    - Netflix uploads versión maestra a la nube de Amazon
    - Amazon crea múltiples versiones de películas (diferentes codificaciones) en la nube.
    - Amazon sube las versiones desde la nube a los CDNs
    - La nube Amazon hospeda las páginas de Netflix
  - *Usa 3 servicios de terceros* para almacenar y distribuir el contenido Netflix: Akamai, Limelight, Level-3 (son 3 CDNs)

# Caso de estudio: Netflix



**DASH: Dynamic, Adaptive Streaming over HTTP**

# Resumen: Multimedia en Internet: varios trucos

- ❑ **use UDP** para abolir control de congestión de TCP (retardo) en tráfico sensible en tiempo
- ❑ **Retardo de reproducción adaptativo en lado del cliente:** para compensar variaciones de retardo
- ❑ Lado servidor **ajusta BW de flujo** a BW disponible en ruta servidor a cliente
  - Elegir entre tasas de flujo pre-codificadas
  - Tasa de codificación dinámica
- ❑ Recuperación de errores (sobre UDP)
  - FEC, entrelazado
  - retransmisiones, si el tiempo lo permite
  - Subsanan errores: repetir datos cercanos
- ❑ **CDN:** traer el contenido más cerca del cliente.

# Capítulo 7: Contenidos

- ❑ 7.1 Aplicaciones Multimedia en Red
- ❑ 7.2 Streaming de Audio y video almacenado
- ❑ 7.3 Real-time Multimedia: Estudio de telefonía en Internet
- ❑ 7.4 Distribución de Multimedia: Redes de distribución de Contenidos
- ❑ 7.5 protocolos para aplicaciones Interactivas de Tiempo Real
  - RTP, RTCP, SIP
- ❑ 7.6 Más allá de Best Effort
- ❑ 7.7 Mecanismos de itineración y políticas
- ❑ 7.8 Servicios Integrados y Servicios Diferenciados
- ❑ 7.9 RSVP