

Introducción a las Redes de Computadoras

Capítulo 2

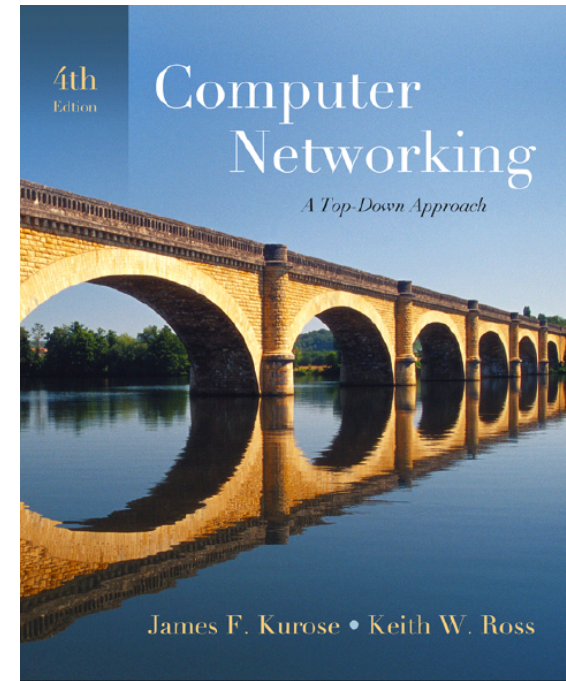
Clase 3

Nota acerca de las transparencias:

Estas transparencias están basadas en el sitio web que acompaña el libro, y han sido modificadas por los docentes del curso.

All material copyright 1996-2007

J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking: A
Top Down Approach,
4th edition.*

Jim Kurose, Keith Ross
Addison-Wesley, July
2007.

Capítulo 2: Capa de aplicación

- 2.1 Principio de aplicaciones de red
- 2.2 Web y HTTP
- 2.3 FTP
- 2.4 Correo electrónico
 - ❖ SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Aplicaciones P2P

DNS: Domain Name System

Personas: como identificarlas

- ❖ CI, nombre, pasaporte

Equipos en internet:

- ❖ Direcciones IP(32 bit)
- ❖ nombre, ej:,
ww.yahoo.com usado por personas

Pregunta: como se relaciona la dirección IP y el nombre?

Domain Name System:

- ❑ *Base de datos distribuída* implementada en una jerarquía de muchos *servidores de nombres*
- ❑ *Protocolo de capa de aplicación* usado para resolver nombres

DNS

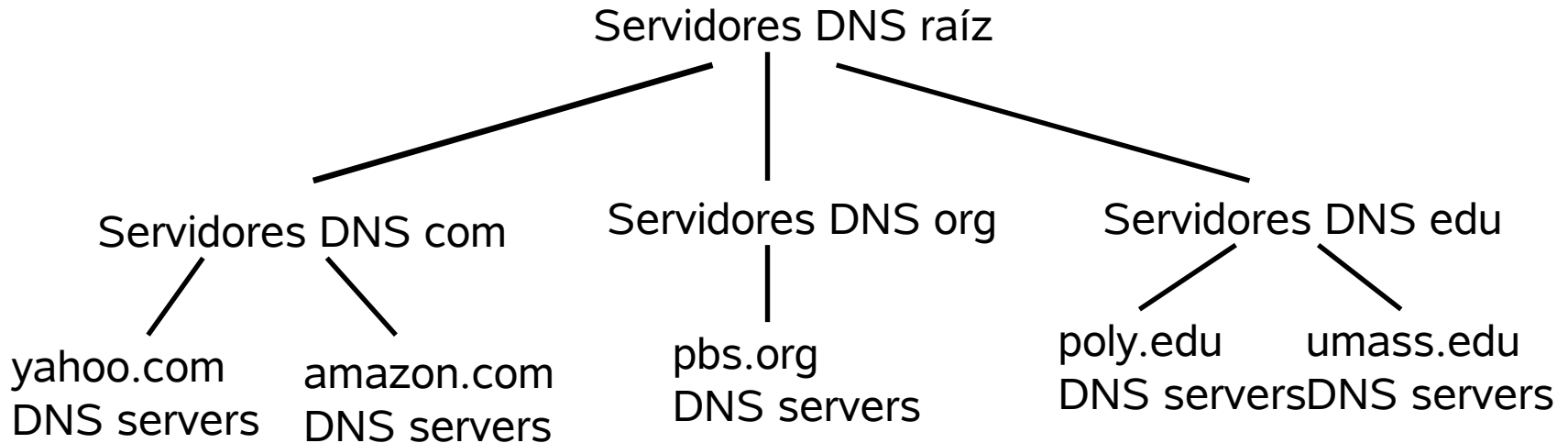
Servicios DNS

- ❑ Traducción de nombre a dirección IP
- ❑ host alias
 - ❖ Canónicos, alias
- ❑ Alias servidor de correo
- ❑ Distribución de carga
 - ❖ Servidores replicados

Por que no DNS centralizado?

- ❑ Punto de falla único
- ❑ Volumen de tráfico
- ❑ Base de datos centralizada distante
- ❑ Mantenimiento
- ❑ NO ESCALA

Base de datos distribuída jerárquica

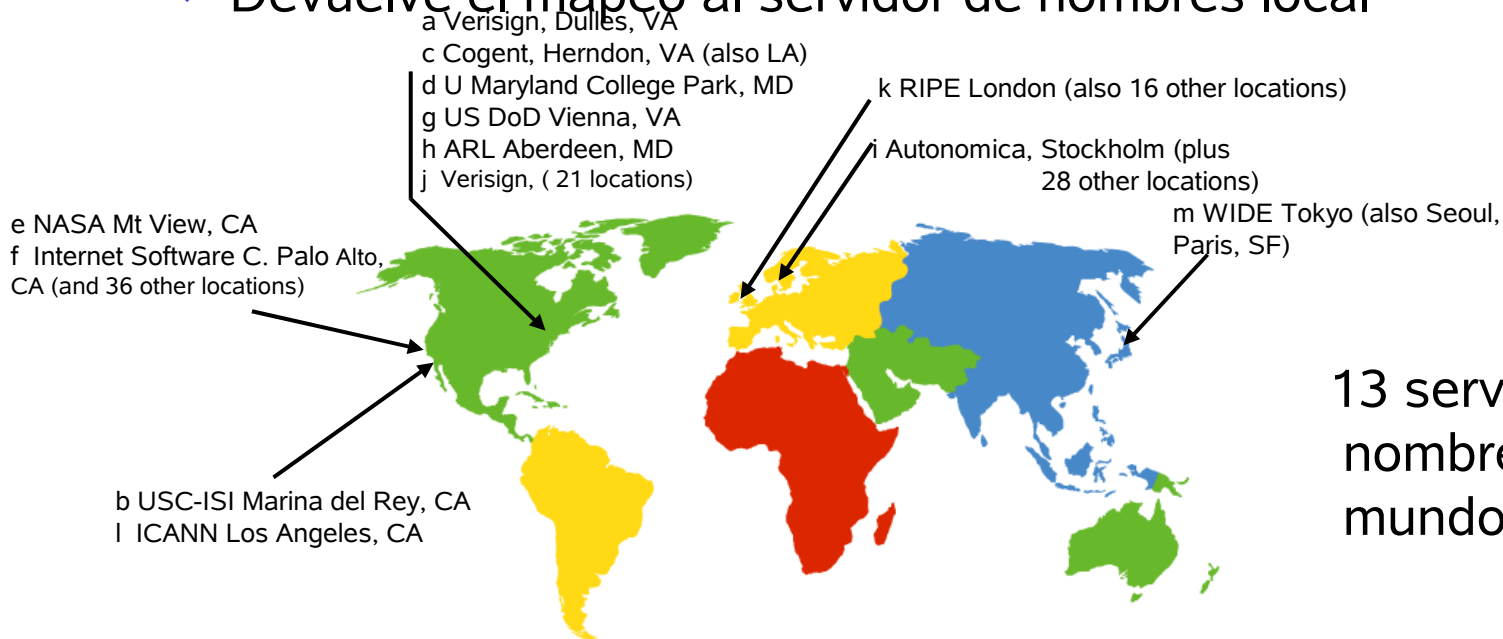


Cliente quiere la dirección IP de www.amazon.com

- ❑ Cliente consulta servidor DNS raíz para obtener servidor DNS com
- ❑ Cliente consulta servidor DNS com para obtener servidor DNS Amazon
- ❑ Cliente consulta servidor DNS amazon para obtener la dirección IP de www.amazon.com

DNS: Servidores raíz

- ❑ Contactados por servidores de nombre locales que no pueden resolver un nombre
- ❑ Servidor de nombres raíz:
 - ❖ Consulta servidor autoritativo si no conoce el mapeo de nombres
 - ❖ Obtiene el mapeo
 - ❖ Devuelve el mapeo al servidor de nombres local



13 servidores de nombres raíz en el mundo

TLD y Servidores autoritativos

□ Servidores Top-level domain (TLD):

- ❖ responsables de com, org, net, edu, etc, y todos los dominios de países uk, fr, ca, jp, uy
- ❖ Network Solutions mantiene los servidores com
- ❖ Educause mantiene los servidores edu

□ Servidores autoritativos:

- ❖ Servidores DNS de organizaciones, proveen mapeos autoritativos (el servidor tiene la autoridad por el registro por el que se consulta, implica que es uno de los servidores del dominio que se consulta)
- ❖ Puede ser mantenido por la organización o el proveedor de servicio

Servidor de nombres local

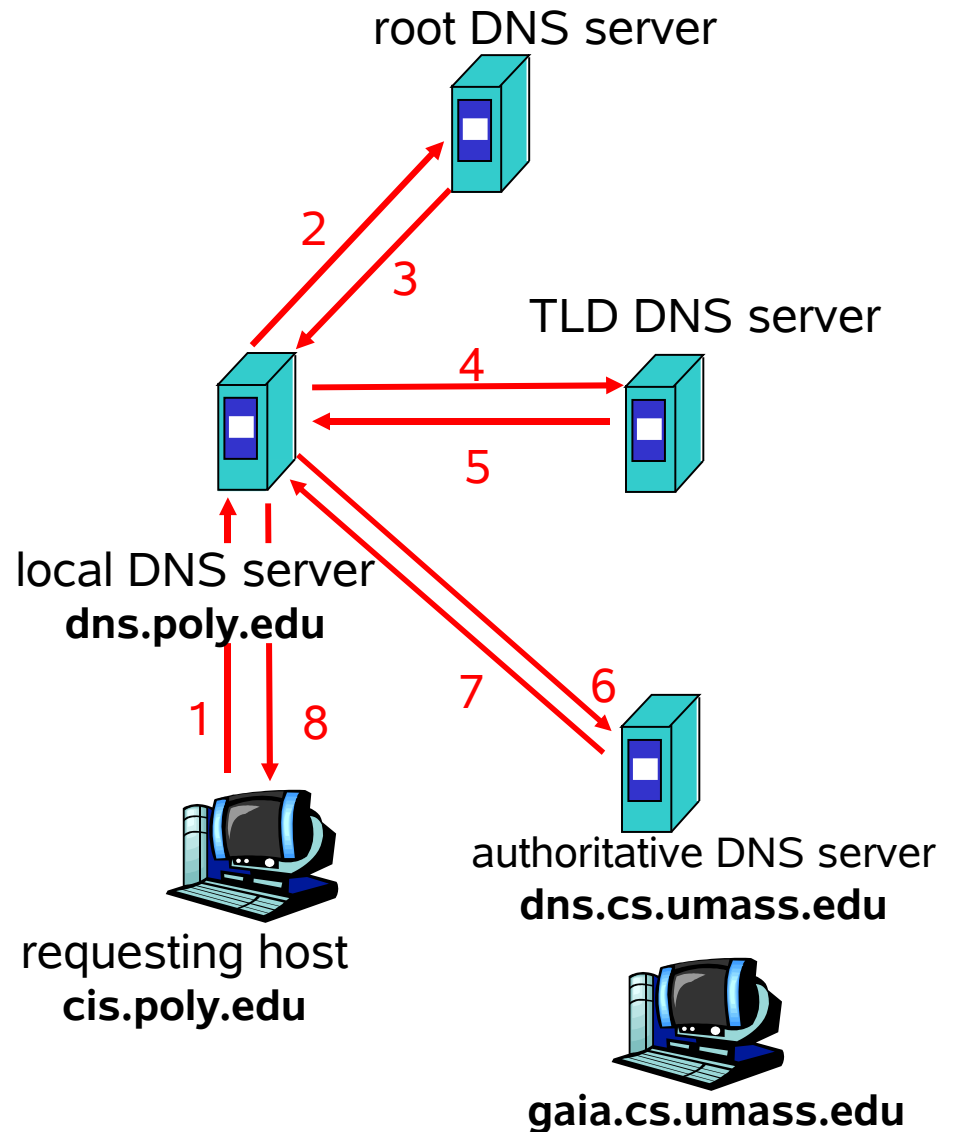
- ❑ No pertenece a la jerarquía
- ❑ Cada ISP (residencial, compañía, universidad) tiene uno.
 - ❖ Se llama “default name server”
- ❑ Cuando un equipo hace una consulta DNS, se envía al servidor local
 - ❖ Actúa como proxy, reenvía la consulta a la jerarquía

DNS ejemplo de resolución

- ❑ Equipo en cis.poly.edu quiere la dirección IP de gaia.cs.umass.edu

Consulta iterativa:

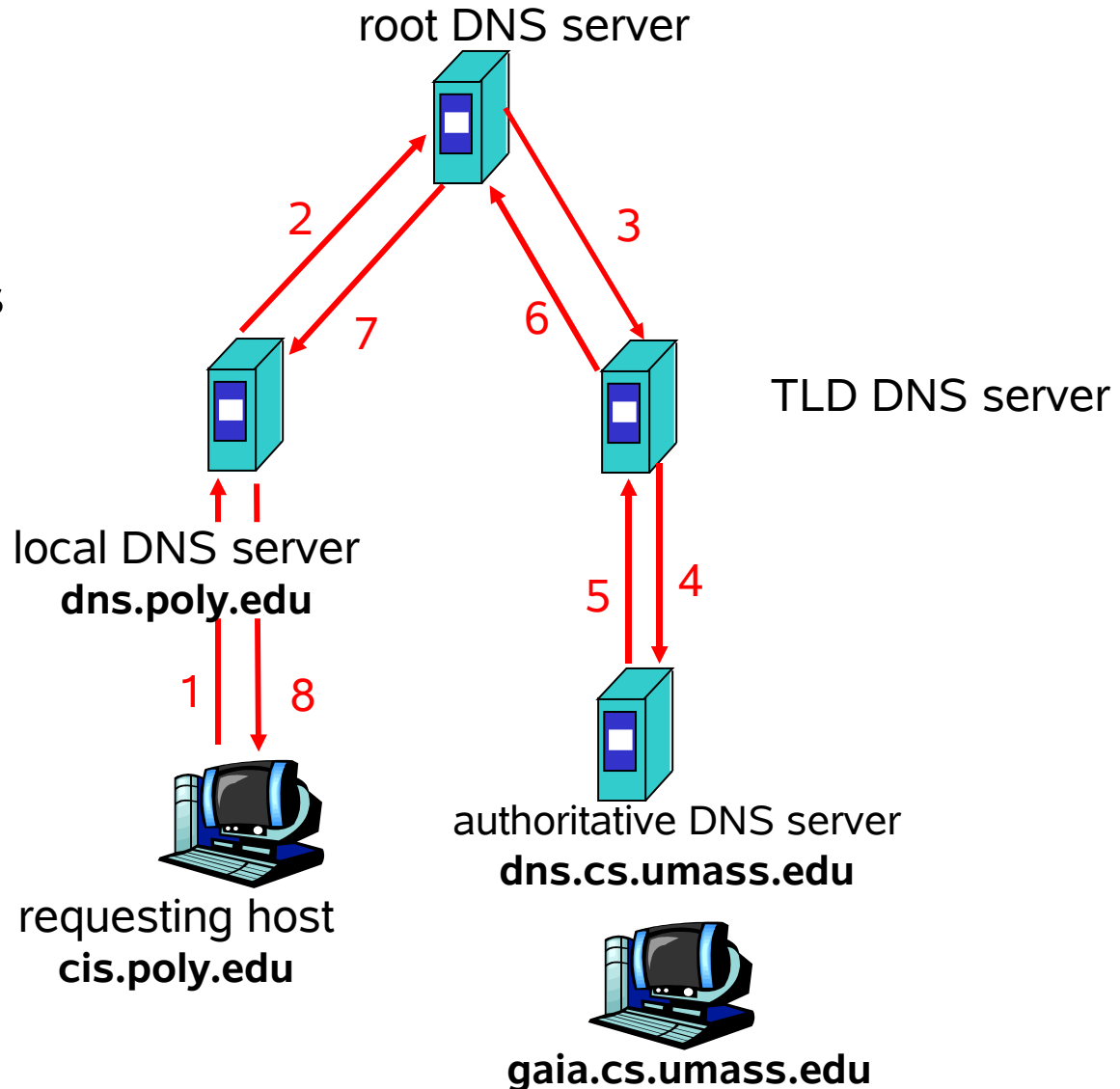
- ❑ El servidor responde con el servidor al cual contactar
- ❑ “No conozco ese nombre pero pregunta a este servidor”



DNS ejemplo de resolución

Consulta recursiva:

- El servidor de nombres se encarga de la resolución



DNS: caching y actualizar registros

- Una vez que el servidor aprende un mapeo, lo cachea
 - ❖ Las entradas vencen despues de cierto tiempo
 - ❖ TLD típicamente cacheados en servidores locales
 - Servidores raíz no son visitados frecuentemente
- Mecanismos de actualización y notificación diseñados por IETF (Internet Engineering Task Force)
 - ❖ RFC 2136
 - ❖ www.ietf.org/proceedings/96dec/charters/dnsind-charter.html

Registros DNS

DNS: base de datos distribuída que almacena los registros de recursos (RR)

Formato RR: (**nombre**, **valor**, **tipo**,
t11)

- Tipo=A
 - ❖ **nombre** es el nombre del equipo
 - ❖ **valor** es la dirección IP
- Tipo=NS
 - ❖ **nombre** es el dominio
 - ❖ **valor** es el nombre del servidor autoritativo para el dominio
- Otros: PTR / SOA / HINFO / TXT / LOC / WKS /SRV / SPF
- Tipo=CNAME
 - ❖ **nombre** es el alias para un nombre “canónico” (real)
`www.ibm.com` es realmente
`servereast.backup2.ibm.com`
Ej:
`dominio=ejemplo.com`
`servicio ftp=ftp.ejemplo.com`
 - ❖ **valor** es el nombre canónico
- Tipo=MX
 - ❖ **valor** es el nombre del servidor de correo asociado con **nombre**

Protocolo DNS, mensajes

Protocolo DNS: consulta y respuesta, mismo formato

Cabezal

- ❑ **identification**: 16 bit
- ❑ **flags**:
 - ❖ Consulta o respuesta
 - ❖ Si se usa recursiva
 - ❖ Si la respuesta es autoritativa

identification	flags
number of questions	number of answer RRs
number of authority RRs	number of additional RRs
questions (variable number of questions)	
answers (variable number of resource records)	
authority (variable number of resource records)	
additional information (variable number of resource records)	



Protocolo DNS protocol, mensajes

Nombre, tipo y campos de una consulta

RRs respuesta a una consulta

Registros para servidores autoritativos usados para seguir con la consulta

Información adicional

identification	flags
number of questions	number of answer RRs
number of authority RRs	number of additional RRs
questions (variable number of questions)	
answers (variable number of resource records)	
authority (variable number of resource records)	
additional information (variable number of resource records)	

↑
12 bytes
↓

Insertar registros en DNS

- Ejemplo: nuevo “Network Utopia”
- Registrar nombre networkutopia.com en *DNS* (ej: Network Solutions)
 - ❖ Proveer nombres, direcciones IP y servidores autoritativos (primario y secundario)
 - ❖ Se insertan dos registros RR en servidor TLD:

(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)

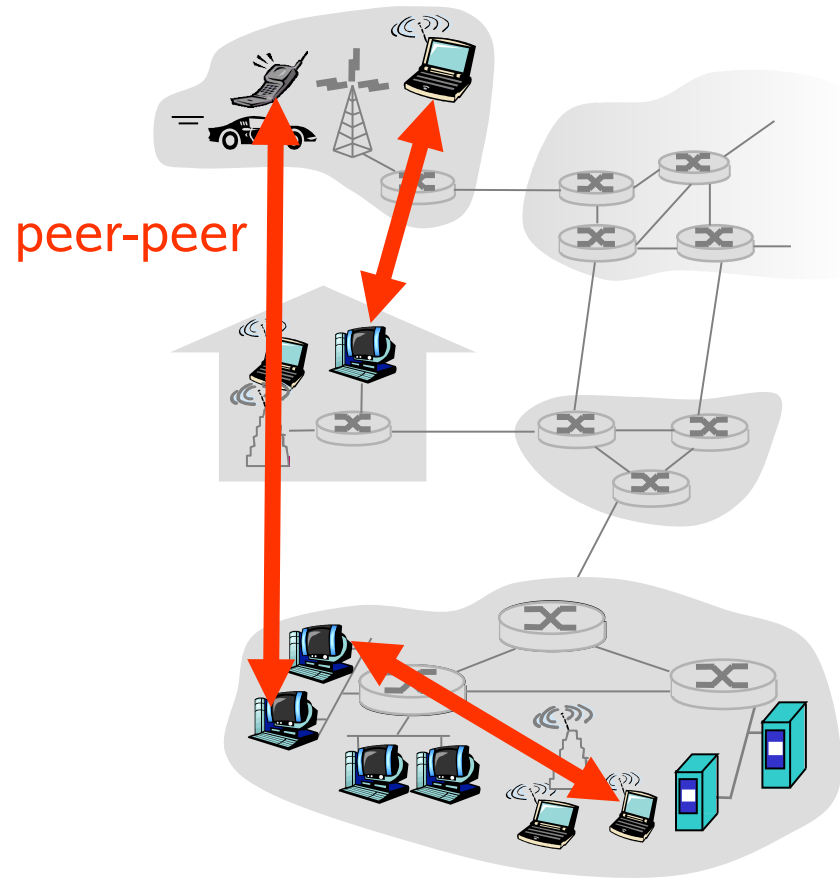
- Crear servidor registro de servidor autoritativo Tipo A para www.networkutopia.com;
Tipo MX networkutopia.com
- Como las personas obtienen la dirección IP del sitio?

Chapter 2: Application layer

- ❑ 2.1 Principio de aplicaciones de red
- ❑ 2.2 Web y HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Correo electrónico
 - ❖ SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 Aplicaciones P2P
- ❑ 2.7 Programación de sockets con TCP
- ❑ 2.8 Programación de sockets con UDP

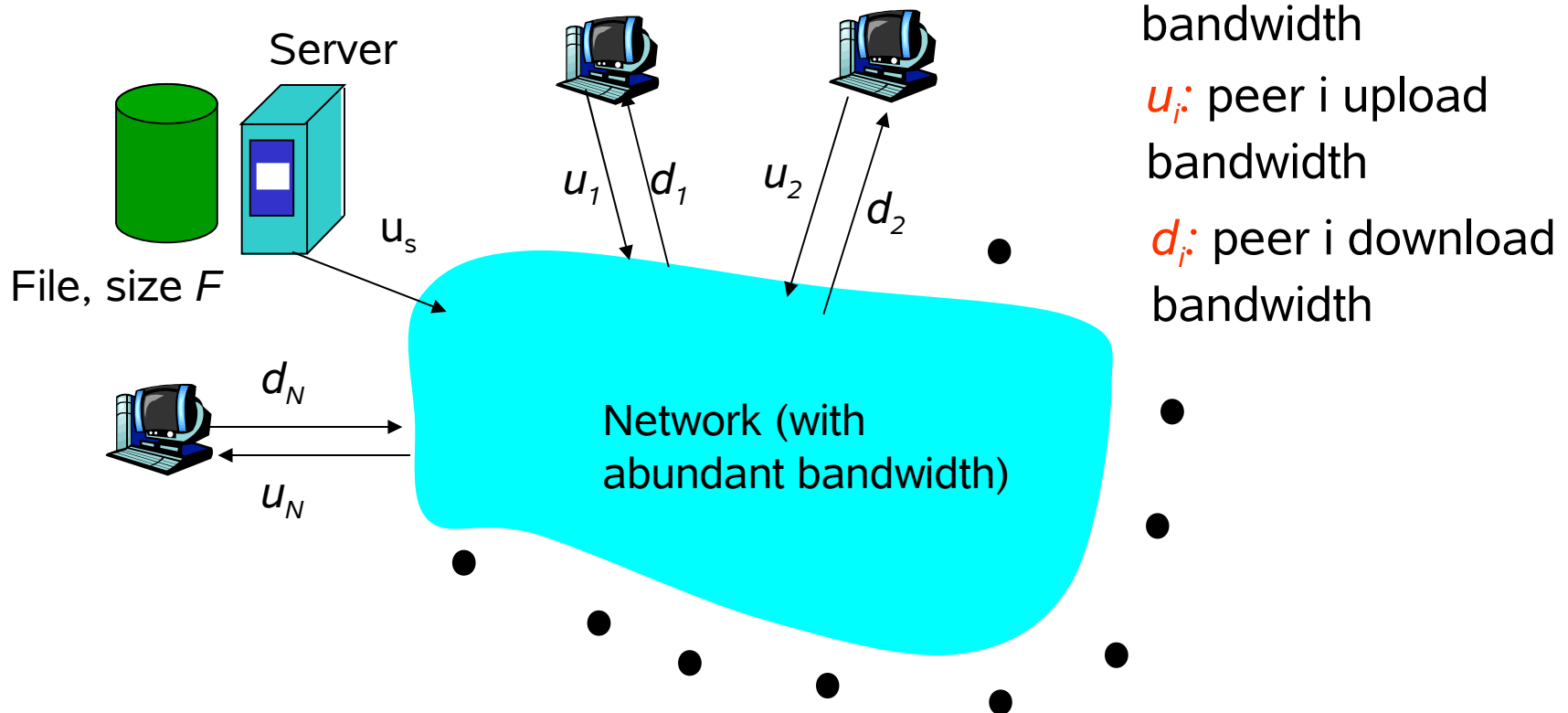
Arquitectura P2P pura

- ❑ *Servidor intermitente*
- ❑ Se comunican sistemas finales variados
- ❑ Cambian direcciones IP
- ❑ Temas de estudio:
 - ❖ Distribución de archivos
 - ❖ Búsqueda de información
 - ❖ Caso: Skype



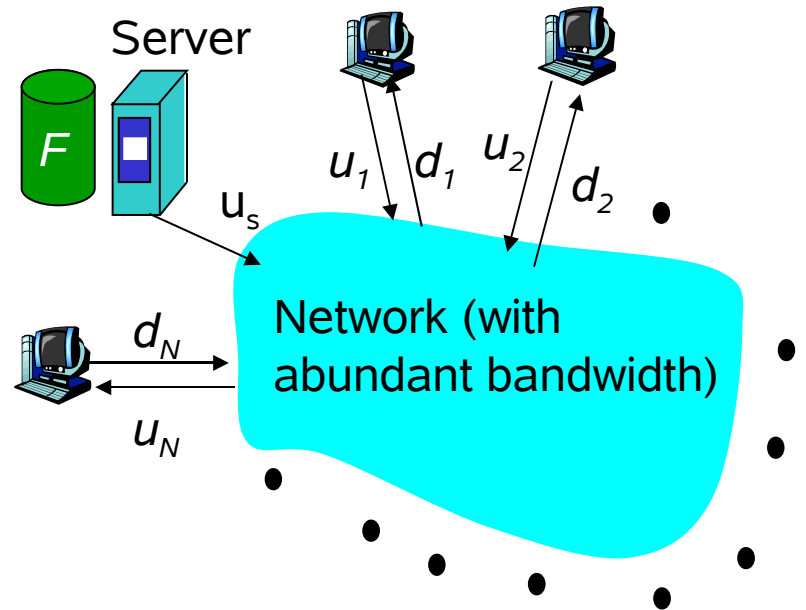
Distribución de archivos: cliente/servidor vs P2P

Cuanto tiempo lleva distribuir un archivo de un servidor a N pares?



Tiempo de distribución de archivo: cliente/servidor

- Servidor envia N copias:
 - ❖ NF/u_s
- cliente i F/d_i para bajar



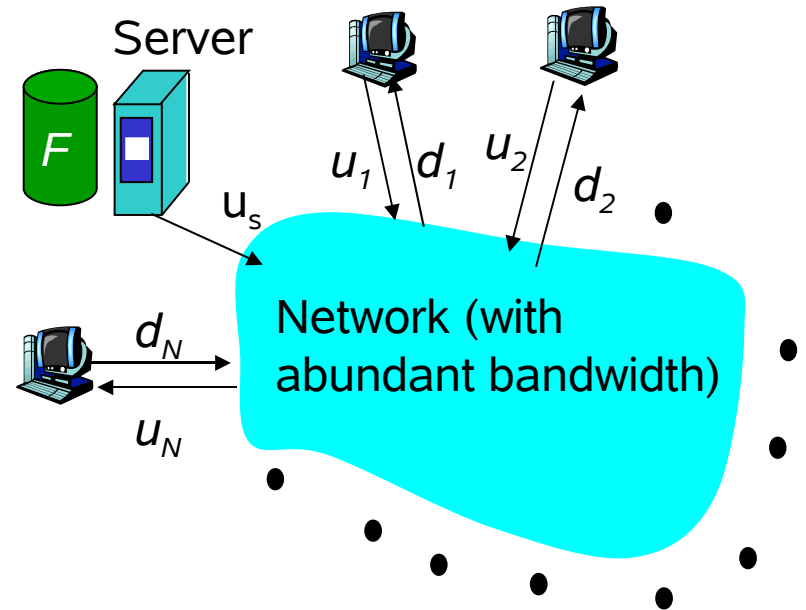
Tiempo para distribuir F

a N clientes $t_{cs} = \max \{ NF/u_s, F/\min_i(d_i) \}$

Incrmenta linealmente en N

Tiempo de distribución de archivo: P2P

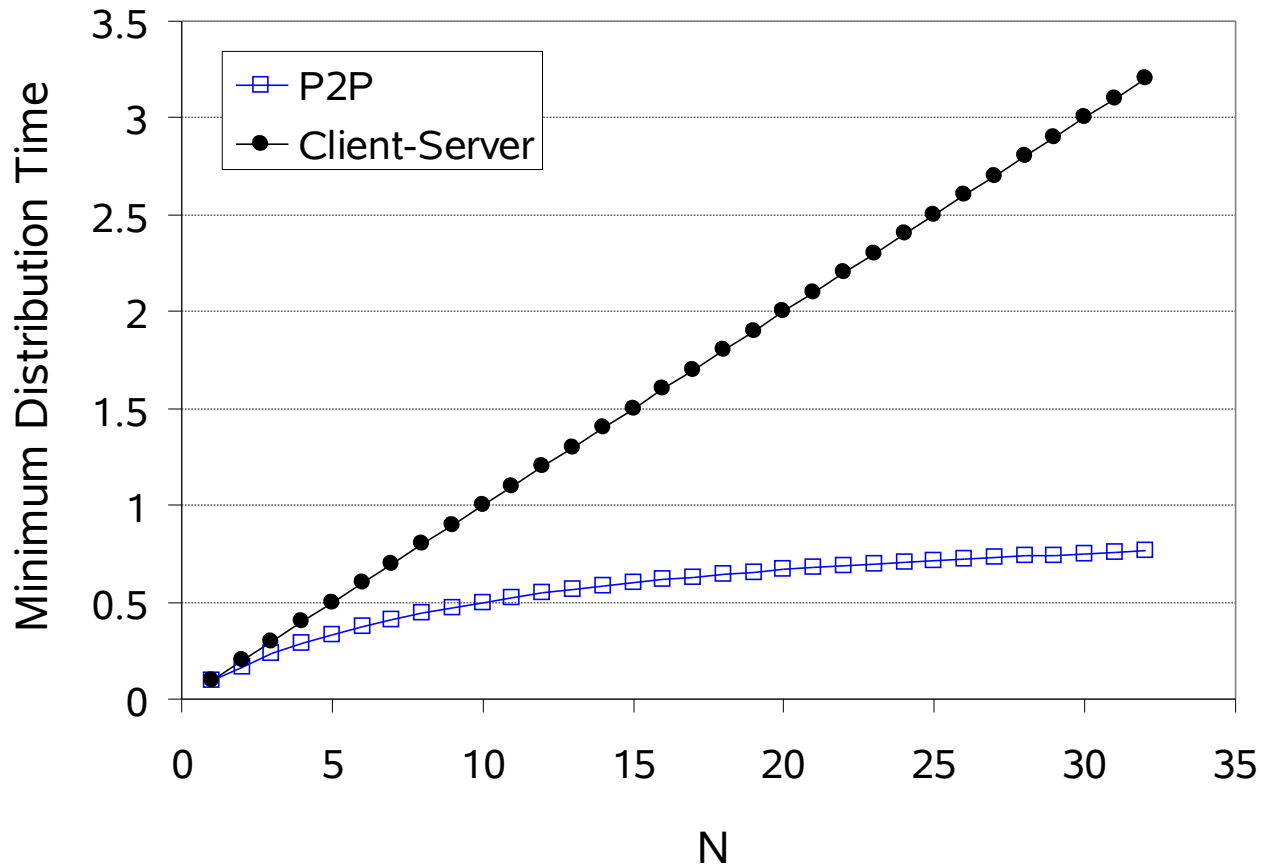
- ❑ Servidor debe enviar 1 copia F/u_s
- ❑ cliente i F/d_i para bajar
- ❑ Se deben bajar NF bits
- ❑ Subida posible: $u_s + \sum u_i$



$$d_{\text{P2P}} = \max \left\{ F/u_s, F/\min(d_i), NF/(u_s + \sum u_i) \right\}$$

Cliente/Servidor vs. P2P: ejemplo

Subida cliente = u , $F/u = 1$ hour, $u_s = 10u$, $d_{\min} \geq u_s$

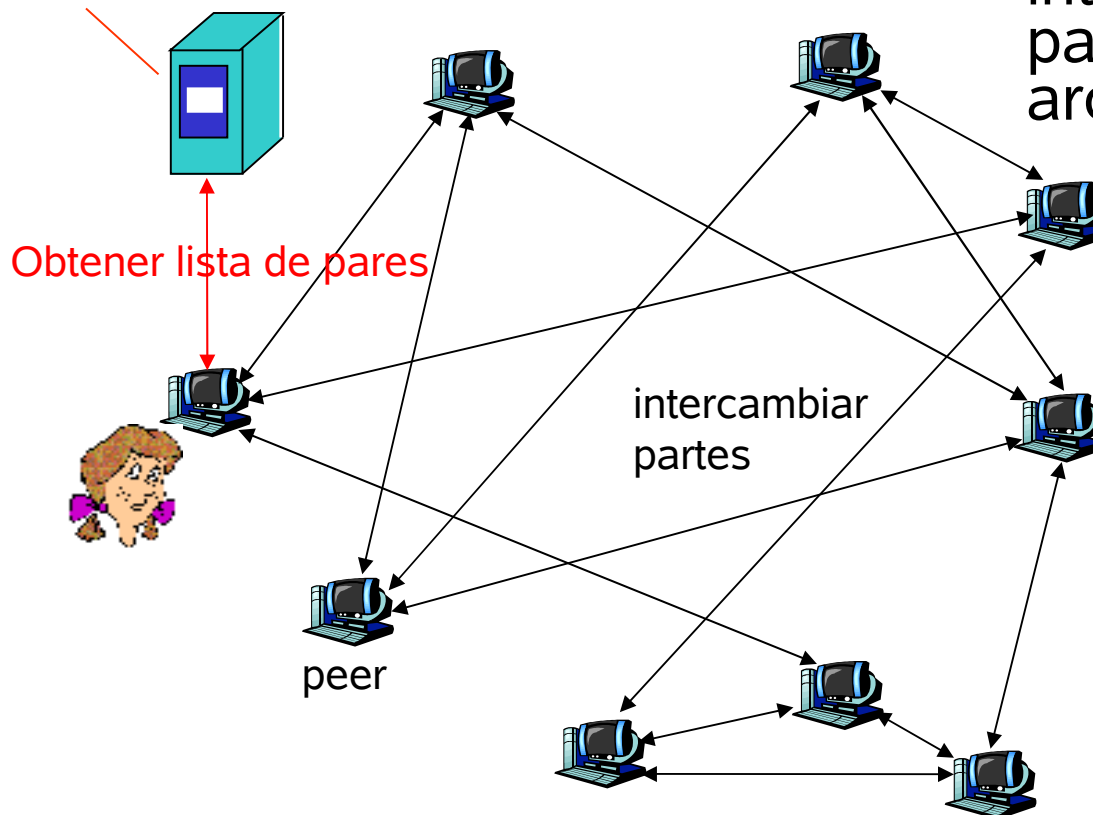


Distribución de archivos: BitTorrent

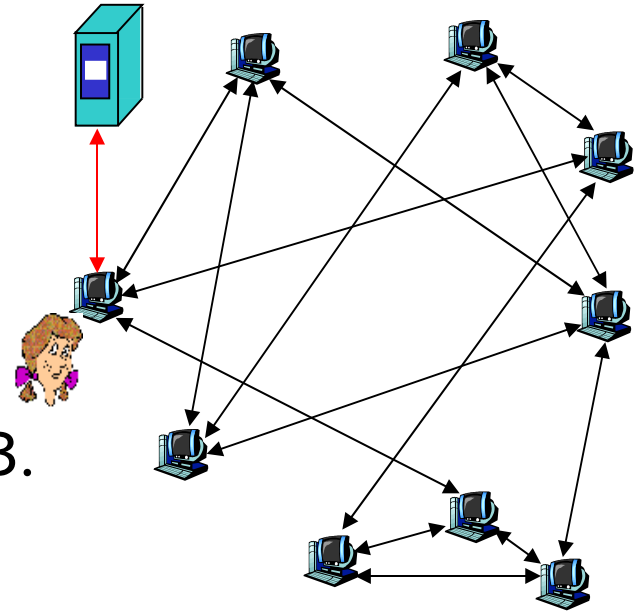
□ Distribución de archivos P2P

tracker: encuentra pares participando en torrent

torrent: grupo de pares intercambiando partes de un archivo



BitTorrent (1)



- ❑ Archivo dividido en partes de 256KB.
- ❑ Par uniéndose al torrent:
 - ❖ No tiene partes, las acumula luego
 - ❖ Se registra con tracker para obtener lista de pares, se conecta a un subconjunto de pares (vecinos)
- ❑ Mientras baja, sube partes a otros pares
- ❑ Los pares se conectan y desconectan
- ❑ Cuando un par tiene un archivo puede desconectarse o seguir conectado para seguir compartiéndolo

BitTorrent (2)

Obtener partes

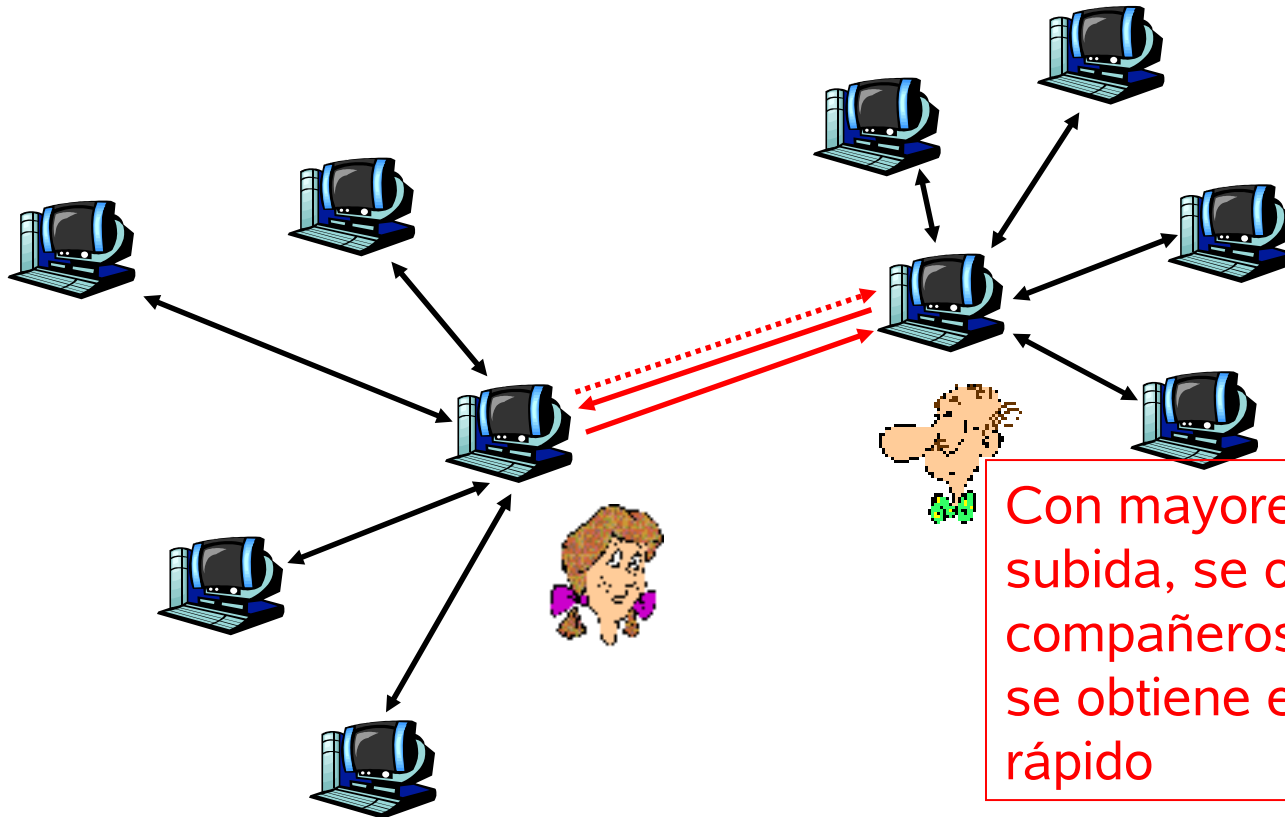
- ❑ En cierto momento, varios pares tienen diferentes conjuntos de partes de un archivo
- ❑ Periódicamente cada par pregunta a sus vecinos por la lista de partes que tienen.
- ❑ Cada par envía pedidos por las partes que le faltan
 - ❖ Primero las partes “raras”

Enviar partes: tit-for-tat

- ❑ Se envían partes a los 4 vecinos enviándole partes a la mayor velocidad
 - ❖ Se reevalúan los 4 cada 10 segundos
- ❑ Cada 30 segundos: se selecciona otro par al azar, se comienza a enviar partes
 - ❖ El nuevo puede unirse a los 4
 - ❖ “desbloqueo optimista”

BitTorrent: Tit-for-tat

- (1) Alicia “desbloquea optimísticamente” a Roberto
- (2) Alicia es uno de los 4 de Roberto; Roberto devuelve el favor
- (3) Roberto se convierte en uno de los 4 de Alicia



Con mayores velocidades de subida, se obtiene mejores compañeros de intercambio y se obtiene el archivo mas rápido

P2P: Búsqueda de información

Índice en un sistema P2P: mapea la información de localización de un par (localización = IP y puerto)

- Compartir archivos
(ej e-mule)

- ❑ Índice dinámicamente ubica los archivos que comparten los pares.
- ❑ Los pares informan que archivos tienen
- ❑ Los pares buscan el índice para ver que archivos pueden obtener

- Mensajería instantánea

- ❑ Índice mapea usuarios y localizaciones
- ❑ Cuando el usuario inicia aplicación, informa al índice su localización
- ❑ Los pares buscan el índice para determinar la dirección del usuario

P2P: índice centralizado

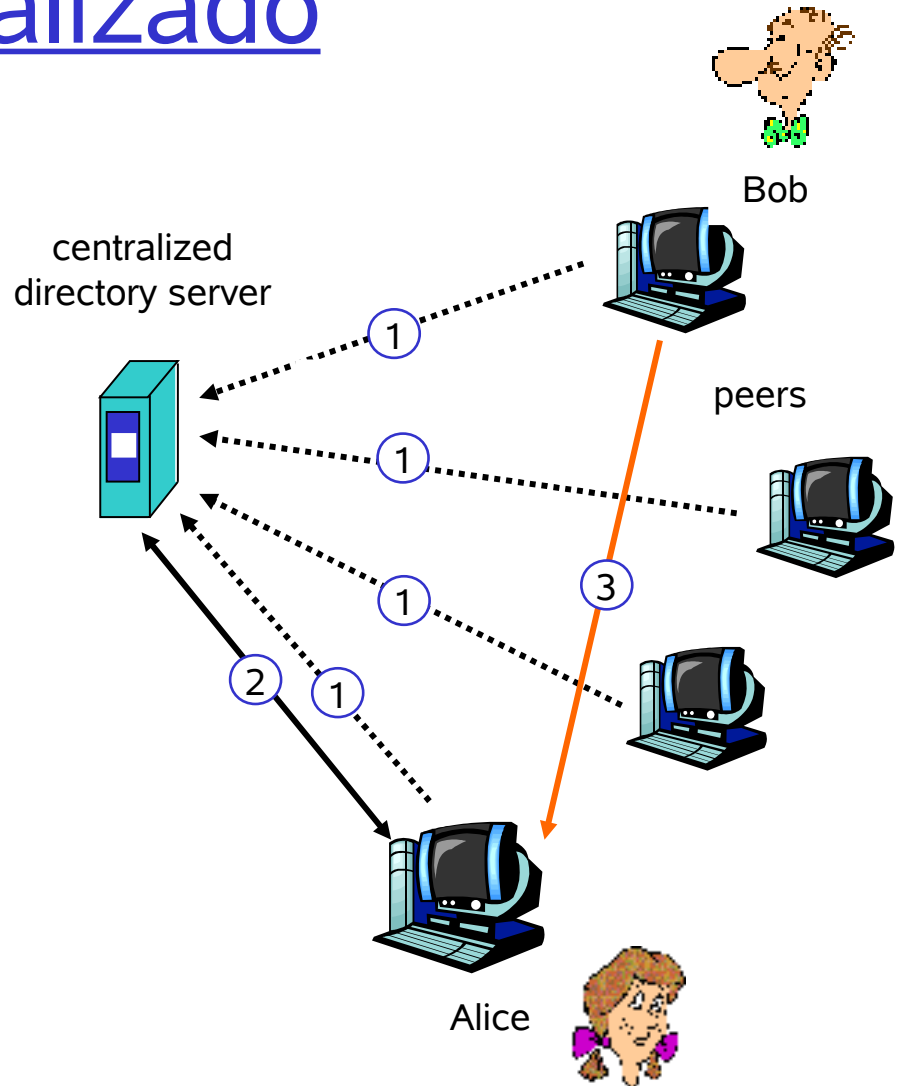
Diseño “Napster”

1) Cuando se conecta cada par, informa al servidor:

- ❖ IP
- ❖ contenido

2) Alicia busca la canción “Yendo a la casa de Damián”

3) Alice pide el archivo a Roberto



P2P: problemas con directorios centralizados

- ❑ Punto de falla único
- ❑ Cuello de botella
- ❑ Violación de copyright

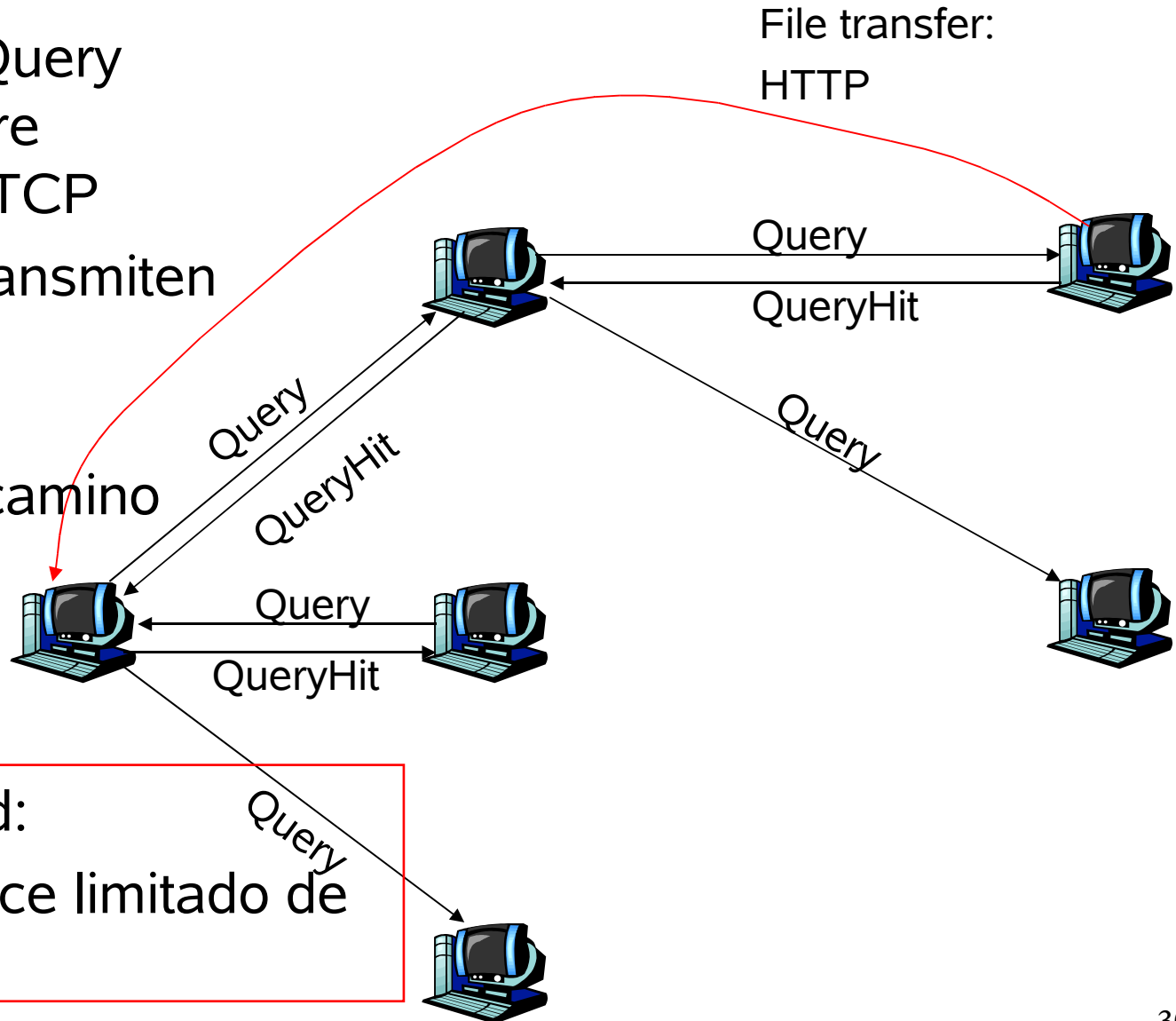
la transferencia de archivos es descentralizada, pero localizar el contenido es centralizado

Inundar consultas

- ❑ Totalmente distribuido
 - ❖ No hay servidor central
- ❑ Usado por Gnutella
- ❑ Cada par indiza los archivos que tiene para compartir

Inundar consultas

- ❑ mensaje Query enviado sobre conexiones TCP
- ❑ Pares retransmiten consulta
- ❑ QueryHit enviado en camino inverso



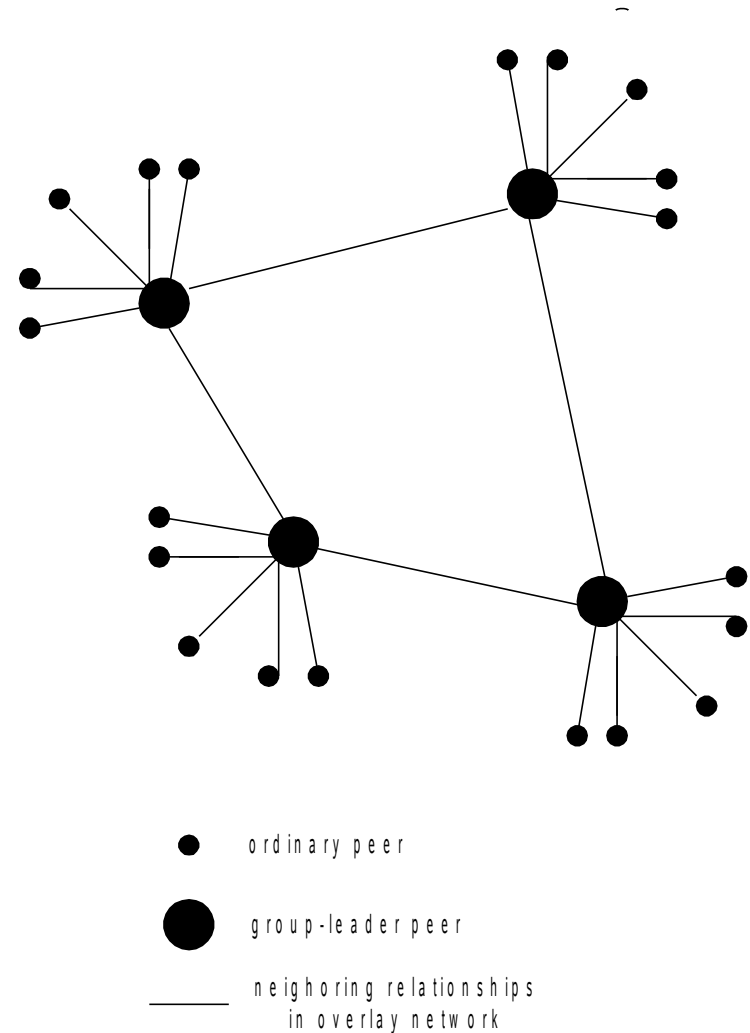
Escalabilidad:
Pobre, alcance limitado de inundación

Gnutella: Unir pares

1. Cuando Alicia quiere unirse a la red debe encontrar otro par Gnutella usando una lista de pares candidatos
2. Alicia intenta conexiones TCP con pares candidatos hasta que logra conectarse con Roberto
3. *Inundación (Flooding)*: Alicia envia Ping a Roberto; Roberto reenvia Ping a sus vecinos y así sucesivamente
 - Los pares que reciben el Ping responden con Pong
4. Alicia recibe muchos Pong, y puede establecer mas conexiones TCP

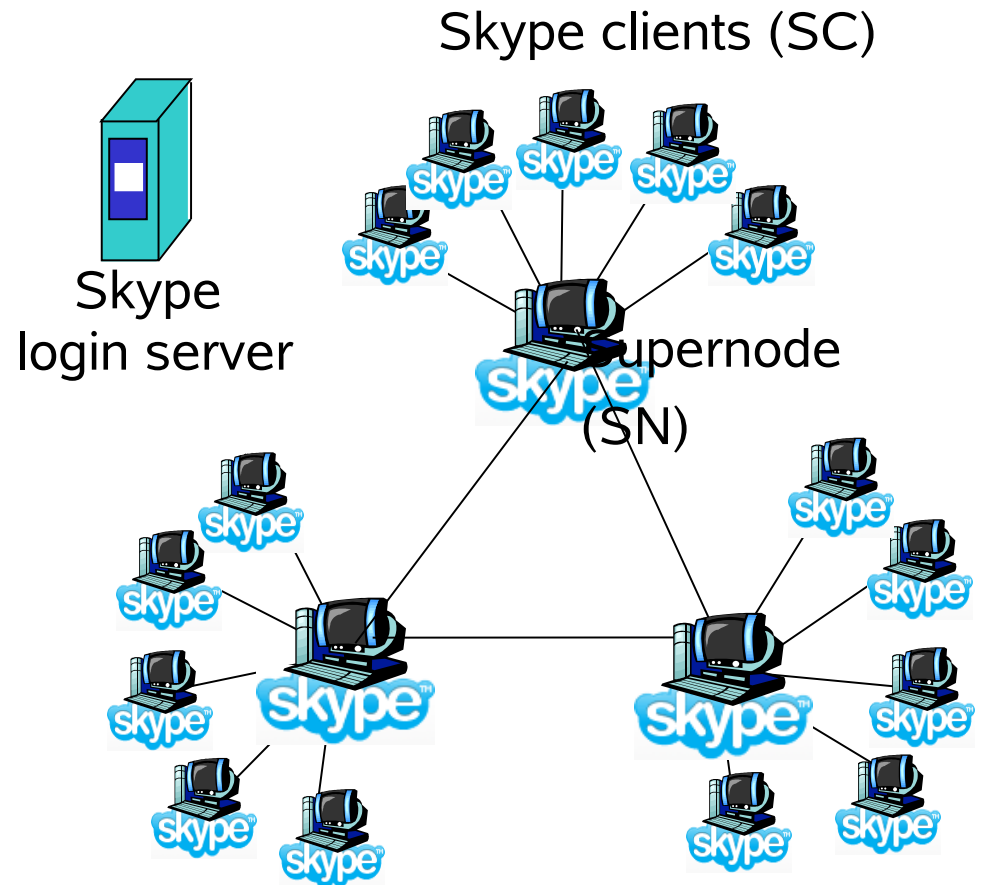
Red de Jerarquía

- Cada par es un supernodo o está asignado a un supernodo
 - ❖ Conexión TCP entre par y su supernodo
 - ❖ Conexiones TCP entre algunos supernodos
- Los supernodos tienen los contenidos de los nodos ordinarios



Caso de estudio P2P: Skype

- inherentemente P2P:
se comunican pares de usuarios
- Protocolo propietario
(inferido por ingeniería
reversa)
- Jerarquía supernodos
- Índice mapea nombres
de usuarios a
direcciones IP,
distribuído en los
supernodos



Pares como relay

- Problema cuando ambos están contra NAT
 - ❖ NAT evita que un par externo inicie una llamada a un par interno
- Solución:
 - ❖ Se utilizan los supernodos de ambos, utilizando relay
 - ❖ Cada par inicia una sesión con relay
 - ❖ Los pares se comunican a través de NAT utilizando relay

