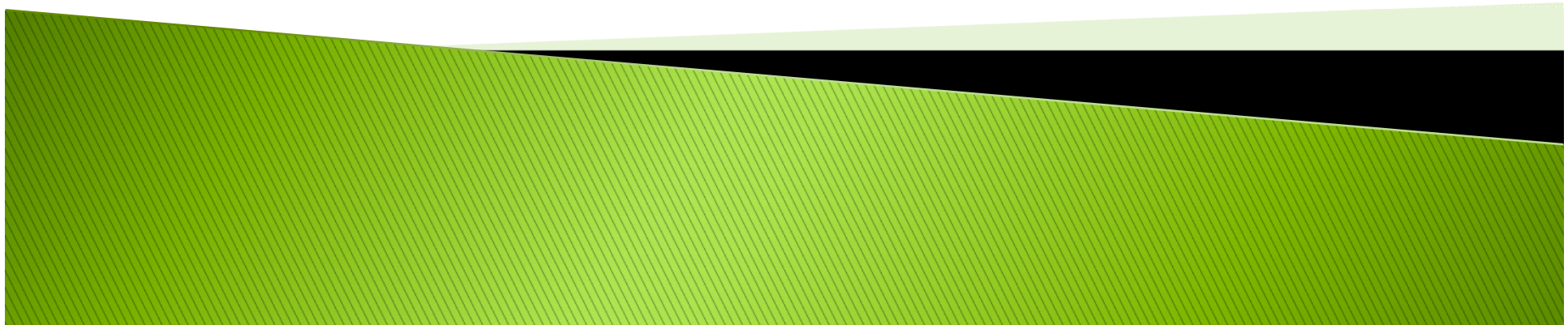




2. Diseño de aplicaciones de comunicaciones

Servicios Telemáticos Avanzados
4º Grado en Ingeniería en Tecnologías de Telecomunicación
Especialidad de Telemática



Indice

Hora 1

- 1 Introducción
- 2 Paradigma cliente/servidor
 - 2.1 Componentes del paradigma cliente/servidor
- 3 Paradigma Peer-to-Peer (P2P)
 - 3.1 Servicios P2P
 - 3.1.1 Localización
 - 3.1.2 Búsqueda
 - 3.1.3 Descarga

Hora 2

- 3.2 Napster
- 3.3 Gnutella
- 3.4 KaZaA
- 3.5 eDonkey
- 3.6 Gnutella2
- 3.7 BitTorrent
 - 3.7.1 Malla de peers
 - 3.7.2 Descarga
 - 3.7.3 Colaboración
 - 3.7.4 Observando un torrente
 - 3.7.5 Siguiendo a un peer
- 4 Conclusiones
- Referencias

1 Introducción

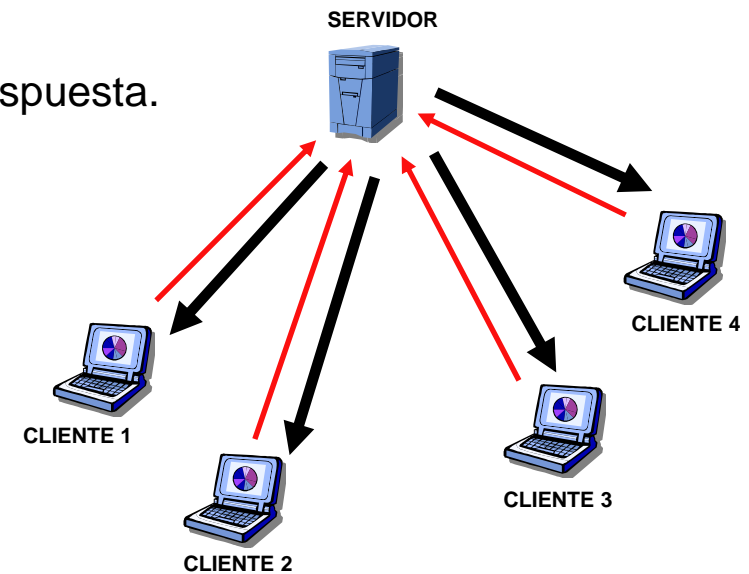
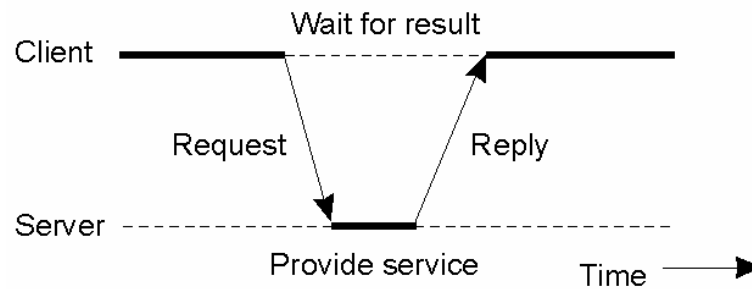
- ▶ Un nivel de red provee un servicio muy simple: reparto de paquetes sin garantías (best effort).
- ▶ El servicio fiable sólo se implementa en un nivel de transporte: TCP
 - Utilizado por multitud de aplicaciones: WWW, Email, FTP, etc.
 - No interfiere con otros niveles de transporte (UDP).
- ▶ Toda otra funcionalidad se implementa a nivel de aplicación.
 - La implementación del servicio es independiente de la tecnologías de red subyacente.
 - No requiere que la red (routers) mantengan ningún estado sobre el tráfico intercambiado por las aplicaciones.
 - Robustez
 - Escalabilidad

Paradigmas de comunicación

- ▶ Cliente/servidor
 - Un nodo servidor atiende a cientos o miles de nodos clientes simultáneamente
- ▶ Peer-to-peer (P2P)
 - Todos los nodos son funcionalmente iguales en la provisión del servicio
- ▶ Híbrido cliente/servidor - P2P
 - Diferentes funcionalidades de la aplicación optan por un paradigma u otro

2 Paradigma cliente/servidor

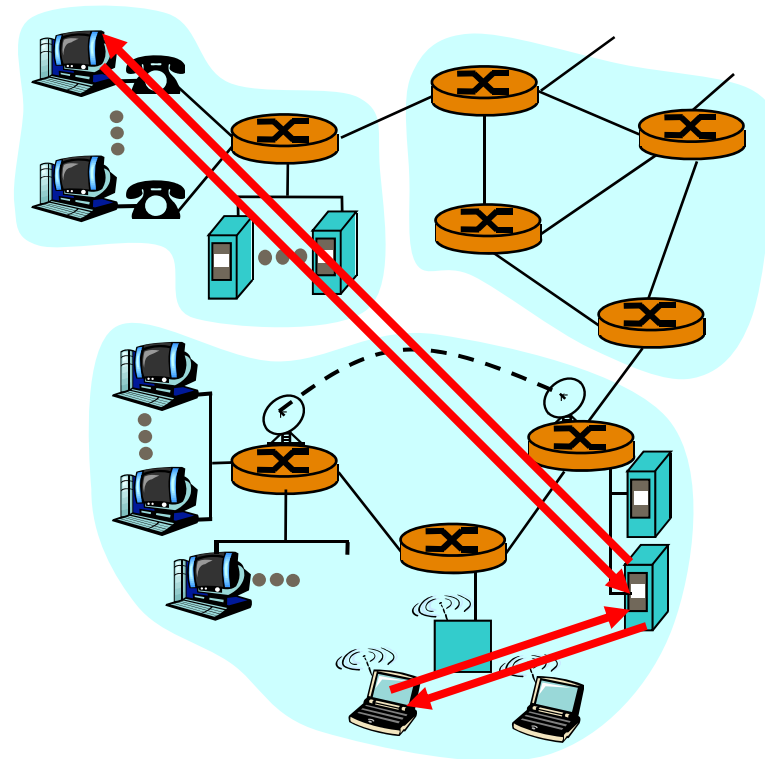
- ▶ **Cliente:** requieren un servicio.
 - Normalmente inicia la comunicación y la finaliza.
- ▶ **Servidor:** prestan un servicio.
 - Normalmente atiende la comunicación iniciada por el cliente.
- ▶ **Patrón típico de petición-respuesta**
 - Cliente: envía solicitud y recibe respuesta.
 - Servidor: recibe solicitud, la procesa y envía respuesta.



- ▶ **Comunicación asimétrica:**
 - Alta carga del servidor
 - Necesidad de elevado ancho de banda en la red de acceso al servidor

Cliente/servidor

- ▶ En la red Internet, habitualmente
 - Cliente
 - Se comunica con uno o varios servidores, no con otros clientes.
 - Actividad intermitente.
 - Dirección IP dinámica.
 - Servidor
 - Máquina siempre activa.
 - Dirección IP fija.
 - Servicio conocido a priori.



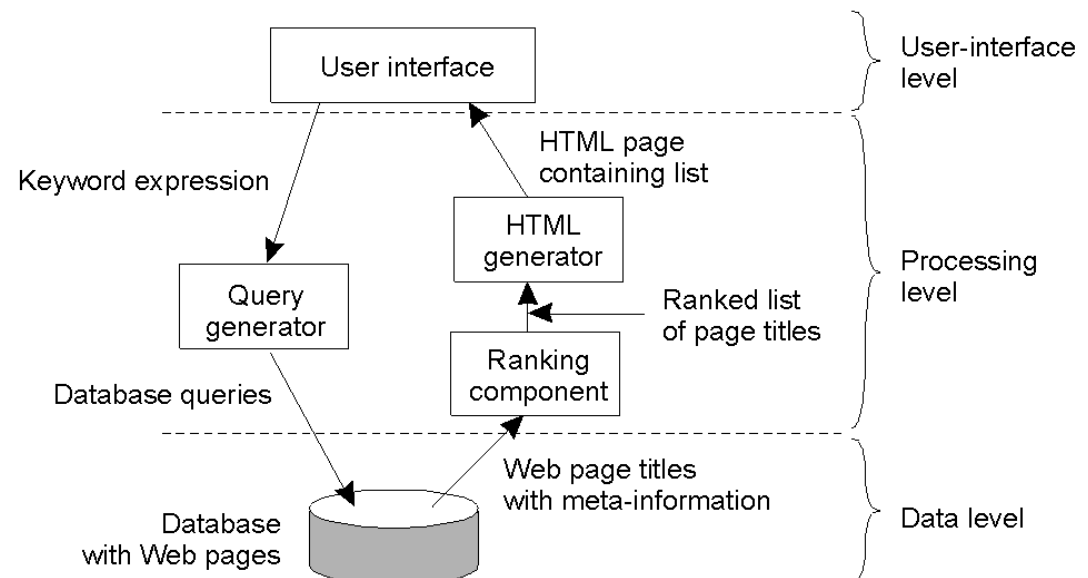
Cliente/servidor sobre nivel de transporte

- ▶ Trabajar directamente sobre el API de sockets disponible en todo sistema operativo:
 - UDP: nivel transporte no fiable orientado a datagramas
 - Menor sobrecarga de protocolo
 - Cabecera básica 8 bytes.
 - Sin mensajes extra de señalización.
 - La aplicación encargada de aportar fiabilidad.
 - Ej: pérdida petición transacción bancaria.
 - TCP: nivel transporte fiable orientado a conexión
 - Mayor sobrecarga de protocolo
 - Cabecera básica 20 bytes.
 - Coste de establecimiento y finalización de la conexión.

2.1 Componentes del paradigma cliente/servidor

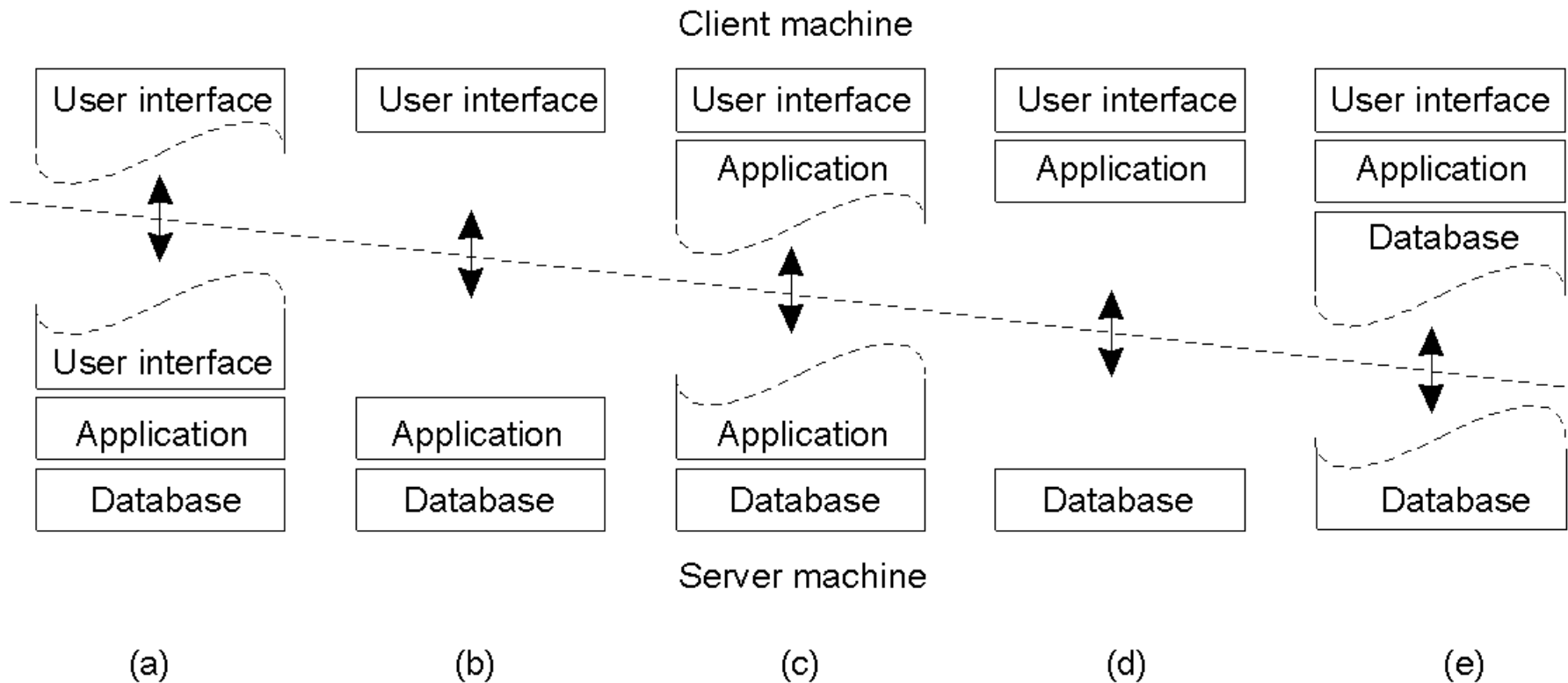
- ▶ 3 niveles o componentes repartidos entre cliente y servidor
 - Interfaz de usuario
 - Desde un interfaz texto a uno gráfico.
 - Procesado
 - Acceso a datos
 - Datos persistentes
 - Ej: base de datos relacional, LDAP, etc.

Ejemplo: motor de búsqueda web



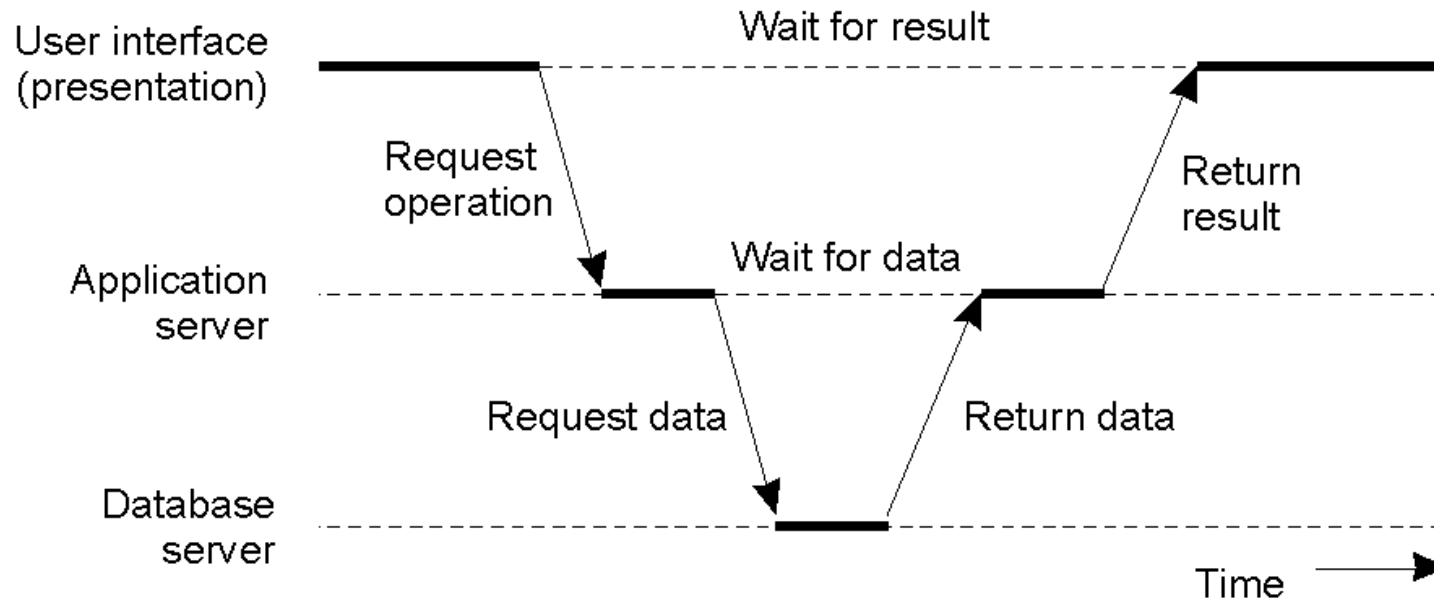
Componentes del paradigma cliente/servidor

- Distribución vertical de los niveles: cada nivel (o parte de nivel) en una máquina física.



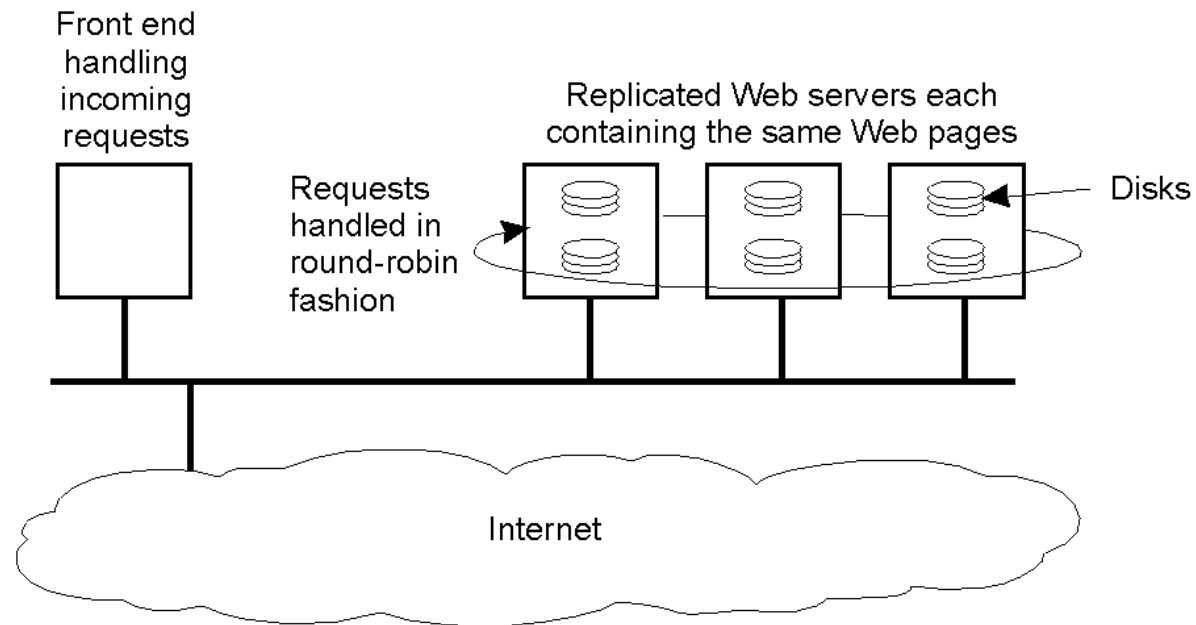
Servidor actuando de cliente

Arquitectura de servicio “three tier”



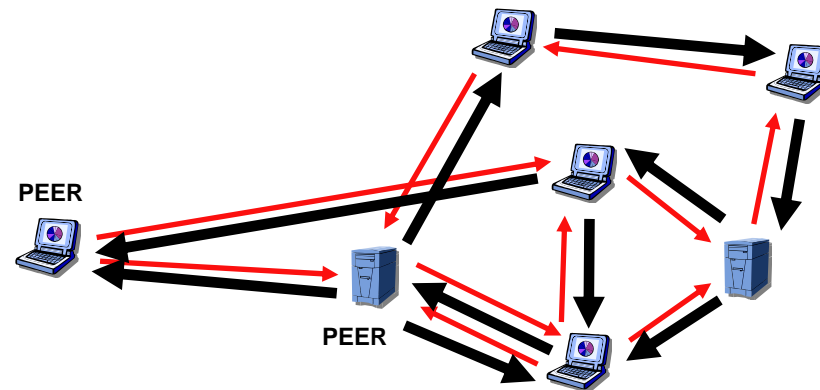
Un cliente y múltiples servidores

- ▶ Distribución horizontal de los niveles:
 - Cada nivel en una máquina lógica (que puede estar distribuida en varias máquinas físicas).
 - Balanceo de carga. Contenidos replicados



3 Paradigma Peer-to-Peer (P2P)

- ▶ Igual-a-igual, par-a-par, todos los nodos son clientes y servidores simultáneamente.
 - Todos los nodos son funcionalmente iguales.
- ▶ Popularidad de aplicaciones de intercambio de ficheros basadas en P2P, superando incluso al tráfico generado por la web.



Ventajas/desventajas P2P

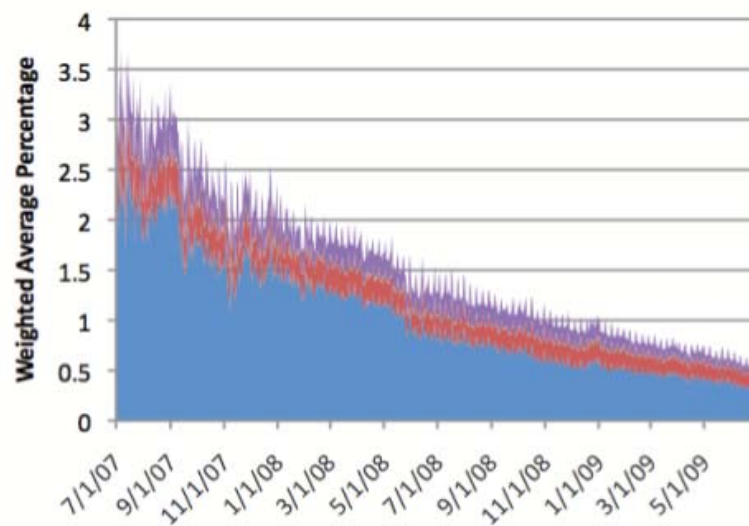
- ▶ **Ventajas del modelo P2P:**
 - Tolerancia a fallos: dado que el contenido está distribuido entre muchos nodos, responde bien ante caídas de algunos de los nodos
 - Escalabilidad: la información se intercambia directamente entre los usuarios finales sin pasar por un servidor intermedio. Se dispone de los recursos (ancho de banda, almacenamiento y CPU) de miles de nodos.
- ▶ **Desventajas del modelo P2P:**
 - Protocolos de complejidad bastante más elevada
 - Consumo de más recursos en cada participante
 - Tiempos de descarga, en general, elevados en comparación con el modelo centralizado (para ficheros distribuidos entre pocos nodos)

Servicios sobre protocolos P2P

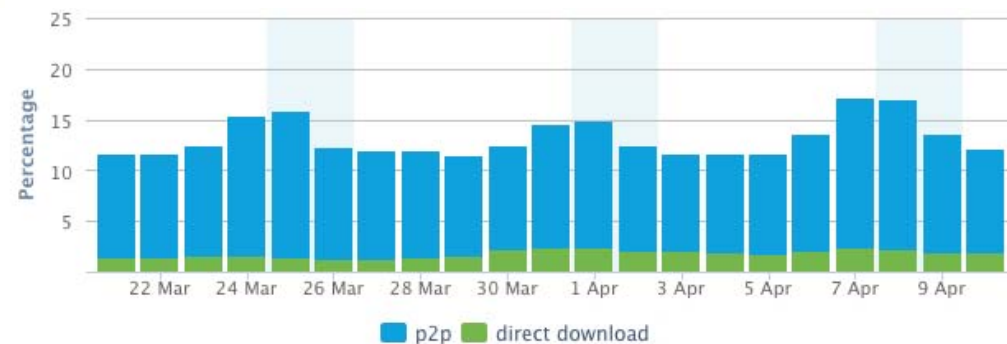
- ▶ Almacenamiento/distribución de contenidos digitales
 - Gnutella, Napster, BitTorrent, etc.
 - Hoy en día la *killer app* del P2P
- ▶ Mensajería instantánea
 - XMPP (Jabber)
- ▶ Videconferencia y VoIP
 - Skype, P2PSIP
- ▶ Computación distribuida
 - seti@home

Importancia tráfico P2P

- ▶ Desde el año 2000, crecimiento continuo del tráfico P2P
- ▶ De 2007 a 2012 caída del tráfico P2P a favor de sistemas de descarga directa
- ▶ En 2012 cierre de Megaupload y de otros servicios de descarga directa han supuesto el resurgir del tráfico P2P

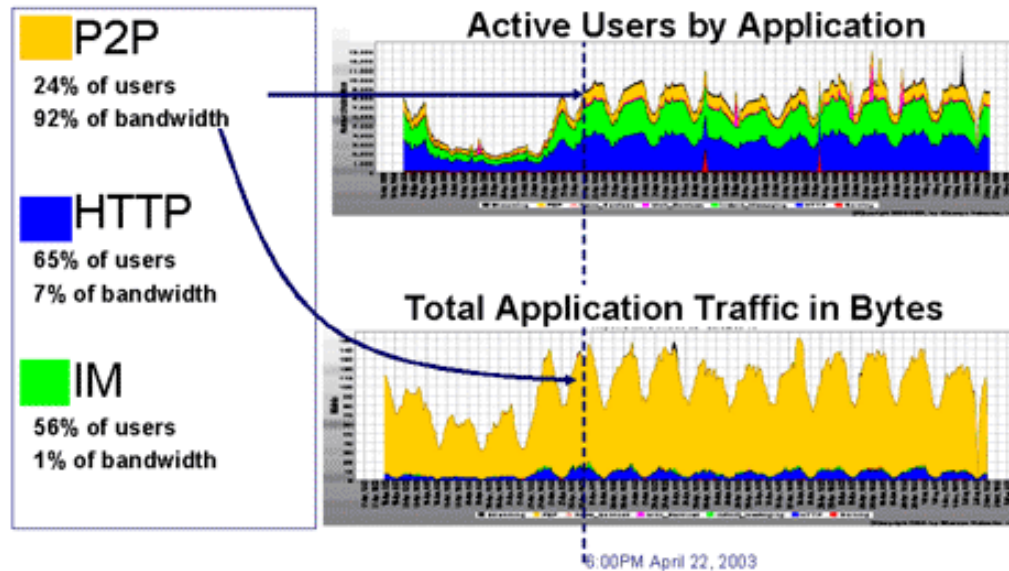


Graph of weighted average traffic using well-known P2P ports



Importancia tráfico P2P

- ▶ Comparado con otros servicios de Internet, las aplicaciones P2P están pensadas normalmente para maximizar la velocidad de descarga



Source: Ellacoya Networks

Organización de la información

- ▶ Diferentes estrategias según la organización de la información:
 - Con servidor centralizado, que registra qué nodo tiene cierto fichero
 - Napster: directorio centralizado
 - BitTorrent
 - Sin servidor centralizado
 - Gnutella: búsqueda por inundación
 - KaZaA: organización jerárquica de la red
 - Híbrido
 - Edonkey/eMule

- ▶ BitTorrent ha disparado el interés por los sistemas de distribución de contenidos peer-to-peer. Otras propuestas prometen mejores resultados [Slurpie] [Avalanche]



3.1 Servicios P2P

- ▶ Problemas a resolver en una red P2P:
 - Localización de otros peers
 - Búsqueda de recursos
 - Descarga de recursos
 - Enrutado de búsquedas y descargas
 - Traspase de NATs
 - Propiedad intelectual, derechos de copyright y privacidad

3.1.1 Localización

- ▶ Para unirse a una red P2P es necesario conocer al menos la dirección IP de un nodo que ya esté en esa red
- ▶ Problemática:
 - Falta de disponibilidad continua de todos los peers
- ▶ Alternativas:
 - Uno o varios servidores centrales donde se registran los nodos en el inicio del cliente P2P, son conocidos a priori y facilitan direcciones IP de nodos conectados
 - Si servidores centrales, necesidad de algún conocimiento de nodos conectados alguna vez al sistema
 - Metafile procedente de un servidor web
 - Histórico mantenido por cada nodo
 - Autodescubrimiento: multicast

3.1.2 Búsqueda

- ▶ Servidor centralizado
- ▶ Sin servidor
 - Inundación
 - Random Walk
 - Distributed Hash Table (DHT)
 - Content Addressable Network (CAN)
 - [...]

3.1.3 Descarga

- ▶ Tipos de transferencias:
 - Tipo napster: de una única fuente.
 - Tipo bittorrent: los peers colaboran entre si (protocolo peer-to-peer) intercambiando partes del fichero



- ▶ Tipos de peers según su actividad:
 - Peers que no colaboran: freeriders, leechers
 - Peers que atacan: polución
- ▶ Colaborar = enviar al otro peer lo que me pide
 - Si no envío, todo el ancho de banda es para recibir
 - Estrategia para que se comparta: si los dos enviamos aun así recibo más que si los dos paramos (créditos, prioridad en colas, etc)

