

Peer to peer (I)

Norberto Fernández, Jesús Arias
Departamento de Ingeniería Telemática

<http://www.it.uc3m.es/berto/>

<http://www.it.uc3m.es/jaf/>



Tecnologías de Distribución de
Contenidos - UC3M



1

Introducción

- Traducción: *de igual a igual* (a veces se traduce como *entre pares*)
- Este nombre transmite la idea de que todos los nodos de la red P2P desempeñan un papel, en esencia, igual
- Esta idea contrasta con la asimetría del modelo cliente/servidor, en que hay una diferenciación clara entre el papel de los nodos cliente y el de los nodos servidor

Tecnologías de Distribución de
Contenidos - UC3M

2

Introducción

- Algunas definiciones alternativas:
 - Clase de aplicaciones que aprovecha los recursos (Ej.: almacenamiento, CPU, contenido, presencia humana) disponibles en los extremos de Internet
 - Sistemas distribuidos no centralizados y auto-organizados en que todas o la mayoría de las comunicaciones son simétricas
 - Excluye algunos sistemas con componentes centralizados

Aplicaciones de los protocolos P2P

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

Distribución de contenidos: Cliente/Servidor vs. P2P

- Modelo centralizado de transferencia de ficheros:
 - El cliente se conecta al servidor y solicita el contenido deseado
 - El servidor envía dicho contenido
- Cuando es necesario, se pueden emplear *pools* de servidores o servidores espejo
- Modelo de, por ejemplo, HTTP y FTP

Distribución de contenidos: Cliente/Servidor vs. P2P

- Modelo P2P de transferencia de ficheros:
 - Cada nodo (*peer*) posee parte del fichero o su totalidad
 - Cada nodo actúa simultáneamente como cliente y servidor:
 - Descarga fragmentos del fichero que aún no posee de otros nodos
 - Proporciona fragmentos del fichero que ya posee a otros nodos
 - Es usual que cada nodo, cuando actúa como servidor, tienda a dar prioridad a aquellos nodos que más contenido le proporcionan
 - Minimizar el *free-riding*

Distribución de contenidos: Cliente/Servidor vs. 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:** un nodo con pocos recursos (CPU, memoria, ancho de banda) puede publicar contenido descargable por una gran cantidad de clientes, dado que una descarga utiliza recursos de muchos nodos

Distribución de contenidos: Cliente/Servidor vs. P2P

- **Desventajas del modelo P2P:**
 - **Protocolos de complejidad bastante más elevada**
 - **Consumo de más recursos en el cliente**
 - **Tiempos de descarga elevados en comparación con el modelo centralizado (para ficheros distribuidos entre pocos nodos)**

Evolución histórica

- 1ª generación: redes centralizadas (Napster)
 - Un servidor central sirve de enlace entre los *peers*
 - Punto central de fallo, sensibles a ataques
 - Problemas de escalabilidad
 - Búsqueda de recursos/*peers* a través del servidor central (rapidez)
 - El servidor central sabe qué recursos comparte cada *peer*

Evolución histórica

- 2ª generación: redes descentralizadas (Gnutella)
 - Objetivo: eliminar el punto central de fallo para evitar el cierre de la red o su caída
 - No existe un servidor central que sirva de punto de contacto de *peers*
 - Todas las comunicaciones son de *peer a peer*
 - Posiblemente a través de otros *peers*
 - Robustez, menor sensibilidad a ataques o caídas de nodos
 - Red carente de estructura

Evolución histórica

- Problemas: búsquedas deben distribuirse a múltiples nodos (no hay un nodo que centralice esa información)
 - Más tráfico en la red
 - Posibilidad de no encontrar lo que buscamos aún cuando esté en la red

Evolución histórica

- 3ª generación
 - Objetivo: Mantener las ventajas derivadas de la descentralización de la 2ª generación minimizando sus inconvenientes
 - Normalmente se introduce cierta estructura en la red para hacer más eficiente la distribución de consultas y/o contenidos. Ideas:
 - Redes con supernodos (Ej.: Gnutella 2, JXTA)
 - DHT (*Distributed Hash Table*) (Ej.: CAN, JXTA, Chord)
 - *Swarming*: Descarga de fragmentos de contenidos de múltiples *peers* a la vez (Ej.: BitTorrent)

Napster

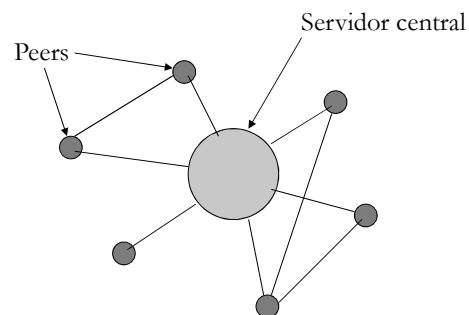
Napster

- Servicio de distribución/compartición de ficheros
 - Fundamentalmente música en MP3
- Aparece a finales del 1999
 - Shawn Fanning, estudiante de la Northeastern University de Boston
- Aplicación que populariza el P2P
 - Más de 26 millones de usuarios en febrero de 2001
- Controversia legal: demandas de discográficas y artistas
 - Cierre por orden judicial en julio de 2001
 - Reaparece más tarde como servicio de pago

Napster

- Red centralizada:
 - Servidor central mantenía lista de *peers* conectados y de qué ficheros compartía cada uno
 - *Peers* buscaban contenidos en el servidor central
 - Una vez encontrado un peer que disponía del contenido de interés, conexión directa entre peers para transferir el archivo deseado

Napster



Gnutella

Gnutella: Introducción

- Protocolo para la búsqueda de recursos distribuidos
- Sistema totalmente descentralizado
- Aparece en el 2000
 - Justin Frankel y Tom Pepper, Nullsoft
 - Actualmente desarrollado por GDF (*Gnutella Developer Forum*)
- Uno de los protocolos P2P más populares con millones de nodos y decenas de aplicaciones software que lo soportan

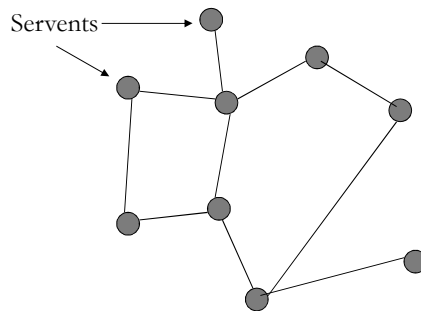
Gnutella

- Nodos son servidores y clientes (*servent*) a la vez:
 - Interfaces de cliente: permiten al usuario realizar consultas y ver los resultados
 - Interfaces de servidor: permiten a otros nodos de la red realizar consultas al nodo
- Una vez se obtienen resultados a la búsqueda, los datos se descargan, si se desea, mediante una conexión directa (Ej.: HTTP) al nodo que los contenga
 - Mecanismo de descarga no especificado por Gnutella

Gnutella

- Un nodo se conecta a la red mediante una conexión TCP a (al menos) un nodo cualquiera que ya forme parte de la misma
 - *Bootstrapping*
- Gnutella no define cómo obtener la dirección de dicho nodo inicial
 - Citarse con un grupo de usuarios utilizando e-mail, mensajería, etc.
 - Existen servidores denominados *host cache* que proporcionan direcciones de nodos de la red

Gnutella



Gnutella: tipos de mensajes

- **Ping:** para descubrir otros nodos de la red
- **Pong:** un nodo que recibe Ping contesta con Pong
- **Query:** mensaje de consulta
- **QueryHit:** un nodo que recibe un mensaje Query contesta con QueryHit, que contiene una lista de resultados relevantes a la consulta
- **Push:** para que nodos que se encuentren detrás de firewalls/NATs y no puedan aceptar conexiones entrantes contribuyan con ficheros a la red

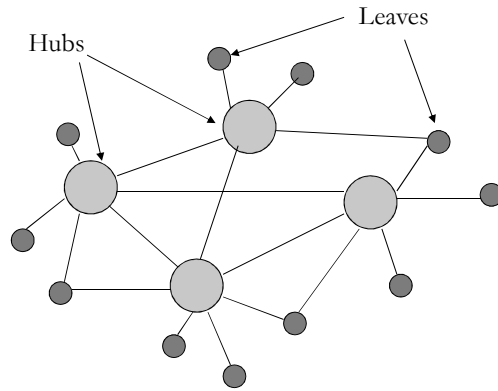
Gnutella: enrutado de mensajes

- **Ping y Query:** un nodo debe reenviar estos mensajes a todos sus nodos directamente conectados, excepto a aquellos de los cuales haya recibido el mensaje
 - Inundación (*flooding*)
 - En cada reenvío se decrementa el TTL del mensaje, hasta que llegue a cero
- **Pong y QueryHit:** se envían sólo por el mismo camino por el que se haya recibido el mensaje Ping o Query correspondiente
- **Push:** se envía sólo por el mismo camino que haya seguido el mensaje QueryHit correspondiente

Gnutella 2

- Gnutella totalmente descentralizado: red robusta ante caídas/desconexión de nodos
 - Cuidado con *host caches*
- Pero problemas de escalabilidad
 - Uso de *flooding* para consultas y ping
 - Mucho tráfico en la red
- Para intentar solventarlos: Gnutella 2
 - Jerarquía de nodos de dos niveles: *hubs* y *leaves*
 - Los *hubs* saben qué ficheros tienen sus *leaves*
 - Las consultas se propagan entre los *hubs*, que son menos

Gnutella 2



Tecnologías de Distribución de
Contenidos - UC3M

25