



Sistemas de Información

Introducción a tecnologías P2P

Agradecimientos: Simon Pickin de IT/UCIIM <spickin>, Jesús Vega Informática/UVA

Índice

- Definición
- Características
- Antecedentes
- Clasificación
- Ejemplos
- JXTA
- Enlaces

Definición de P2P

- **Peer:**

- otro entidad del mismo nivel

- **Peer-to-peer communication:**

- comunicación de-igual-a-igual

- **Peer-to-peer network:**

- redes de pares / redes de iguales
- no depende de servidores dedicados; utiliza (al menos mayormente) conexiones directas entres nodos participantes
 - una red P2P pura no tiene clientes y servidores; todos los nodos son del mismo nivel
 - una red puede denominarse P2P a pesar de que utilizan una estructura cliente-servidor para ciertas tareas
- cada nodo tiene capacidades y responsabilidades equivalentes
- cualquier nodo puede iniciar una interacción con otro nodo
- distintos nodos pueden tener distinta velocidad de procesamiento, distinto ancho de banda y distinta cantidad de almacenamiento

Algunas características de P2P

■ Descentralización

- devuelve el poder y autonomía a los nodos en el borde de la red (edges)
- maneja conexiones variables y direcciones provisionales
- inteligencia distribuida
- fenómeno social
 - democratización?

■ Aprovecha el ancho de banda de los usuarios

- utilizado completamente
- ancho de banda total crece con el número de nodos
- los clientes no tienen que compartir el ancho de banda limitado de un servidor

■ Destaca la **distinción** entre publicar y ser autor

- *NOTA: no confundir “par” en su significado de “igual” con “par” en su significado de “pareja”; por otra parte, nos referiremos aquí a cada nodo de una red P2P como “un par”.*

Caracterización de P2P

- Características claves según Dave Winer (XML-RPC)
 - las interfaces de usuario cargan fuera de un navegador Web
 - los ordenadores de los usuarios pueden actuar como clientes y como servidores
 - el sistema global es fácil de usar y bien integrado
 - el sistema incluye herramientas para soportar los usuarios que quieren crear contenido o añadir funcionalidad
 - el sistema proporciona conexiones con otros usuarios
 - el sistema hace algo nuevo o emocionante
 - el sistema soporta protocolos multi-red tal como SOAP o XML-RPC

Antecedentes: Usenet news

- Concebido en 1979 estudiantes de Duke University
- Sistema de intercambio de noticias. Copia ficheros entre máquinas distintas
- Los usuarios leen o envían mensajes denominados artículos a distintos grupos de noticias organizados jerárquicamente
- Modelo descentralizado de control. (en la práctica, hay un *backbone*).
- Protocolo de base
 - original: UUCP (*Unix-to-Unix Copy Protocol*)
 - actual: NNTP (*Network News Transport Protocol*) sobre TCP/IP
 - optimizaciones, p.e. encabezado de camino (*path header*) para evitar mensajes repetidos

Antecedentes: Usenet news

■ Funcionalidad:

□ Para unirse

- un servidor de *news* monta una conexión de intercambio de *news* con uno o varios otros servidores de *news*

□ Una vez adscrito

- un servidor de *news* conecta regularmente con su vecino / sus vecinos
- intercambian los mensajes nuevos

□ Creación de un nuevo grupo de *news*

- se propone, se discute y se vota en el grupo **news.admin**

Antecedentes: DNS

- *Solución al problema de compartir archivos*
- *Domain Name System*
 - red P2P junto con un modelo jerárquico de propiedad de información
- Escalabilidad
 - 1983: unos miles de nodos
 - 2005: cientos de millones de nodos (~10.000 veces más grande)
- Modelo híbrido:
 - Filosofía P2P en las comunicaciones
 - Método jerárquico de propiedad de la información. Delegación de responsabilidad según la jerarquía de nombres
- Nodos pueden ser clientes (delegan petición) y servidores (cachean respuestas)
- Un servidor puede preguntar a cualquier otro pero
 - camino estándar para propagar peticiones por la cadena de autoridad
- Distribución de carga
 - cada servidor de nombres sólo se ocupa de su espacio de nombres

Antecedentes: DNS

■ Funcionalidad:

- Petición: un host quiere conocer la dirección IP partiendo del nombre
 - Consulta a su servidor de nombres más próximo
 - Si ese servidor no lo sabe delega la consulta a la autoridad de ese espacio de nombres
 - Esto se repite a través de toda la jerarquía hasta llegar a la solución
- La respuesta
 - se propaga en la jerarquía en sentido contrario
 - se cachea en todos los nodos intermedios para que la respuesta sea más eficiente en siguientes peticiones

Clasificación arquitectural de redes P2P para compartir ficheros

■ Centralizada

- se mantiene un directorio puesto al día constantemente en localizaciones centrales
- un nodo hace peticiones al directorio para encontrar los nodos que contienen los ficheros deseados
- directorio central es punto de fallo crítico

■ Descentralizada y estructurada

- no hay directorio central
- la topología de la red está controlada y los ficheros se colocan en localizaciones que facilitan peticiones subsiguientes

■ Descentralizada y no estructurada

- no hay directorio central
- la topología de la red y la colocación de los ficheros no están controladas
- primeros sistemas, tráfico excesivo
 - búsquedas por *flooding* de peticiones basado en TTL

Clasificación generacional de redes P2P para compartir ficheros

■ Primera generación

- utiliza una lista de ficheros centralizada, p.e. Napster
- desde 2001: quien controla esta lista central es responsable de cualquier violación de copyright u otra actividad ilegal que pueda ocurrir (p.e. intercambio de ficheros pornográficos etc.)

■ Segunda generación

- utiliza una lista de ficheros descentralizada, p.e. Gnutella
- evita el problema de responsabilidad legal de unos por las acciones de otros

■ Tercera generación

- utiliza una lista de ficheros descentralizada, p.e. GUNet
- mejoras en cuanto a eficiencia, fiabilidad, anonimato, etc.

Algunos ejemplos de redes P2P: ICQ (I seek you)

- Desarrollado por Mirabillis 1996 adquirido por AOL 1998
- 1er Servicio completo de mensajería instantánea
- Analogía: Conversación
- Permite chatear, envío de mensajes, intercambio de archivos, etc.
- El usuario se le asigna un UIN (Universal Internet Number) en el momento de registrarse
- Protocolo utilizado: Oscar (Open System for CommunicAtion in Realtime).
 - Es propietario
 - Descifrado por ingeniería inversa e implementado por competidores que han desarrollado clientes para ICQ (Microsoft y Jabber)

Algunos ejemplos de redes P2P: Jabber (Conversational technologies)

- Surge en 1998 como un intento de crear una plataforma abierta distribuida para mensajería instantánea para evitar incompatibilidad entre servicios IM existentes.
- Muchos participantes individuales y comerciales.
- <http://www.jabber.org>
- Proyecto opensource que combina mensajería instantánea con XML (streaming XML)
- Creación espontánea de comunidades de usuarios estructurando la información que les interesa

Algunos ejemplos de redes P2P: Jabber (Conversational technologies)

- Conjunto de tecnologías
 - Un servidor modular escrito en C
 - Clientes para casi todas las plataformas (comerciales y openSource)
 - Gateways para la mayoría de los servicios y protocolos de mensajería en internet
 - Bibliotecas para multitud de lenguajes de programación
 - Agentes y servicios especializados como traductores y RSS
- Objetivo mejorar las capacidades conversacionales entre personas y entre aplicaciones

Algunos ejemplos de redes P2P: Jabber (Conversational technologies)

- Uso de XML para casi todo:
 - Codificación de los datos
 - Protocolo de transporte entre peers
 - Formato de almacenamiento
 - Modelo interno de datos de la mayoría de las aplicaciones
- Soporte namespaces para que el usuario decida sobre las etiquetas XML a utilizar
- El protocolo Jabber ha sido aceptado por el IETF bajo el nombre XMPP (**E**Xtensible **M**essaging and **P**resence **P**rotocol) con RFC 3920

Algunos ejemplos de redes P2P: Jabber (Conversational technologies)

■ Funcionamiento

- Identidad (3 partes): usuario@servidor/recurso
- Desde fuera se accede como: jabber://user@server/resource/data
- Similar al e-mail. El cliente se conecta al servidor que es el responsable de negociar la recepción y entrega de los datos con otros usando los protocolos disponibles. Servers son peers para los servers y clientes para los clientes
- Es protocol-agnostic. Traslada a XML los datos de un protocolo a otro de forma transparente para el usuario
- Capacidades similares a cualquier sistema de mensajería
 - Información de presencia (en documento XML)
 - Listas de contactos
 - En versiones recientes navegación entre listas de contactos

■ Ejemplo: Google talk (agosto 2005)

- Usa combinación de VoIP e IM usa Jabber para mensajería instantánea
- Desde 17 de Enero 2006 soporta comunicación entre servidores

Algunos ejemplos de redes P2P: SETI@home

- Red de computación distribuida
- Analogía: CPU distribuida
- SETI = Search for Extra-Terrestrial Intelligence
 - analizar señales de radio que provienen del espacio
 - buscar patrones que podrían indicar fuentes no naturales
 - 50GB de datos por analizar al día
- Aprovecha el tiempo libre de miles de PCs
 - datos divididos en unidades de trabajo por frecuencia y por tiempo
 - cada unidad analizado en la máquina de un voluntario

Algunos ejemplos de redes P2P: SETI@home

■ Funcionamiento: el cliente

- se conecta con el servidor, obtiene una unidad de trabajo y cierra la conexión
- analiza los datos
- se reconecta con el servidor, entrega los resultados, recoge otra unidad de trabajo y cierra la conexión

Algunos ejemplos de redes P2P:

Napster

- Una red para compartir de ficheros
- Inicio: mediados de 1999; autor: Shawn Fanning (19 años)
- Primer demanda judicial: finales de 1999
- Software bajado por ~30 millones de personas en 12 meses
- Cierre de Napster dictado por el juez en julio 2001
 - en ese momento, número medio de usuarios simultáneos: ~160 mil

Algunos ejemplos de redes P2P: Napster

■ **Funcionamiento:**

- los pares se conectan con el servidor central y proporcionan
 - su dirección IP
 - el contenido que ofrecen
- un par envía la petición de un fichero al servidor
- el servidor proporciona los datos de los pares que lo están ofreciendo
- el peticionario elige entre ellos (*pings* para buscar el más rápido)
- el peticionario recupera el fichero

Algunos ejemplos de redes P2P:

Gnutella

- Un protocolo de software libre
 - Objetivo sistema de búsqueda distribuido
 - encaminamiento al nivel de aplicación
 - basado en mensajes, no en circuitos
 - existen muchas aplicaciones distintas que utilizan el mismo protocolo
- Desarrollado al principio del 2000
- Sistema de búsqueda y descubrimiento de recursos distribuido.
- Analogía: Cocktail party

Algunos ejemplos de redes P2P: Gnutella

■ Funcionamiento

- un par se conecta con un nodo Gnutella
- su mensaje está difundido entre los otros nodos Gnutella locales
- todos estos vecinos contestan
- un par envía la petición de un fichero a todos sus vecinos
- estos la re-envían a sus vecinos; de este modo, la petición se propaga por la red (*flooding*) en un radio determinado por el TTL
- las respuestas se encaminan hacia el peticionario
- el peticionario elige una respuesta y se conecta con el nodo en cuestión para obtener el fichero

Algunos ejemplos de redes P2P: Freenet

- Una red para compartir ficheros y ancho de banda
- Ian Clarke de la universidad de Edimburgo
- Analogía: Disco duro enorme, geográficamente distribuido y con acceso rápido y anónimo
- Sistema descentralizado objetivos
 - Prevenir la censura
 - Garantizar anonimato de los usuarios
 - Eliminar puntos de fallo o de control
 - Eficiencia en almacenamiento y distribución
- Sistema de nodos que intercambian mensajes.
 - Dos tipos de mensajes insertar o eliminar.
 - Los mensajes se identifican con id
 - Sistema de almacenamiento: clave-datos-dirección

Algunos ejemplos de redes P2P: Freenet

■ Funcionamiento (petición)

- A cada petición se le asigna UID
- Nodo envía consulta con identificador del recurso a buscar
 - Normalmente inicia consulta a un nodo que conoce y en quien confía (normalmente nodo corriendo en su misma máquina)
 - Si no encuentra en BD local busca en su tabla de enrutamiento el nodo más eficiente para localización de este recurso
 - Inicialmente cuando no se conoce desempeño de otros nodos se hace aleatoriamente
 - Cada petición enrutada mediante saltos entre vecinos hasta que se encuentra el doc o se rebasa n° de saltos permitidos
 - Cada vecino que no tiene el doc pasa la petición y recuerda que lo hizo.

Algunos ejemplos de redes P2P: Freenet

■ **Funcionamiento** (respuesta)

- La respuesta viaja por el camino inverso que siguió la petición.
- Los nodos intermedios deciden si cachean o no el documento.
Cachearlo:
 - Mejora futuras consultas los documentos más buscados están en más nodos
 - Dificulta localización de las fuentes de cara a la censura
- La respuesta contiene la dirección de uno de los nodos a través de los cuales viajó para mantener la red cada vez mejor conectada
 - Cada nodo no conoce la cadena a través de la cual le llegó la petición, sólo conoce a su vecino más próximo es difícil saber quien hizo la petición
 - En realidad sí se pueden hacer averiguaciones pero el método sería poco eficaz

Algunos ejemplos de redes P2P: Freenet

■ Funcionamiento (inserción)

- Antes de insertar documento se envía solicitud para localizarlo
 - Si se encuentra devuelve el recurso
 - No es necesaria la insercción
 - Evita ataques que traten de insertar masivamente un documento
 - Si no se encuentra
 - No se envía mensaje de timeout
 - Se envía el documento por la misma ruta que viajó la petición y todos los nodos intermedios lo cachean
 - De este modo nos aseguramos que el documento está en los nodos en los que se va a consultar

Algunos ejemplos de redes P2P: Freenet

- Funcionamiento (almacenamiento)
 - Pila donde cada clave se asocia a los datos y la dirección del nodo del que proceden
 - Los documentos que reciben muchas peticiones
 - se almacenan en local
 - se mueven hacia la cima de la pila
 - La distancia que se mueven depende de su tamaño (esto fomenta que la gente introduzca documentos comprimidos)

Edonkey 2000

- Red basada en el protocolo del mismo nombre.
 - Una de las redes más populares
 - Su nombre viene del primer cliente, actualmente muy poco usado
 - Cerrado en Septiembre de 2005
- Funcionamiento:
 - Uso de servidores para interconectar clientes.
 - División de ficheros en partes para ficheros > 9,28 Mb cada una
 - Cada parte tiene su propio hash (MD4)
 - Identificación del hash del fichero completo, para identificarlo de forma única independientemente del nombre que tenga.
 - Uso de enlaces, que conducen a un único archivo gracias al hash (MD4) que lo tiene identificado sin ambigüedad, lo que evita el problema de archivos falsos.
 - La división de archivos en partes permite compartir el fichero antes de que esté completo

BitTorrent

- Una red para compartir ficheros de gran tamaño optimizando uso de ancho de banda
- Creado por Bram Cohen. Estrenado en 2002 y Escrita en Python y publicado bajo la licencia MIT
- Objetivo: Mecanismo eficiente para distribuir un mismo fichero a un gran número de personas.
- Funcionamiento similar a eDonkey2000
 - eDonkey uso más ineficiente del ancho de banda porque usuarios eDonkey comparten y descargan máyor cantidad de ficheros
 - eDonkey no premia a los que comparten mayor ancho de banda

BitTorrent

■ Funcionamiento

- Divide ficheros en fragmentos de 256KB
- Para iniciar transferencia se crea un fichero .torrent y se abre en el cliente.
- El fichero contiene:
 - nombre del fichero a transferir
 - Tamaño
 - código hash de cada fragmento
 - dirección de un tracker (mantiene nombres de peers y seeds en un swarm) no involucrado en transmisión de fichero
 - Información para el usuario
- Existe un peer que tiene una copia completa del fichero a transferir y hace de seed (semilla). Su papel como seed termina cuando ha distribuido todos los fragmentos

BitTorrent

■ Funcionamiento

- Los peers que quieren descargar un mismo archivo forman un enjambre (swarm).
- Los peers de un mismo swarm reciben algún fragmento del seed y luego intercambian entre ellos las piezas que les faltan
- Para mejorar eficacia
 - Se elige el peer con el que existe mejor conexión
 - Se empieza descargando los fragmentos más raros (disponibles en menor nº de peers) de este modo evitan cuellos de botella
- Los fragmentos no se descargan secuencialmente
- Cada peer empieza a servir fragmentos desde que ha descargado el primer fragmento completo

Justificación JXTA

■ Proyecto JXTA:

- Necesidad de crear aplicaciones P2P sólidas, sobre una base bien definida y común.
 - Contexto: Aplicaciones P2P actuales con protocolos propietarios
 - Objetivo: Interoperabilidad y reutilización
- Proyecto JXTA de Sun (*Juxtapose*, contraponer)
 - Permite implementar soluciones P2P sobre cualquier dispositivo con “corazón digital”, PDAs, Móviles, ...
 - Independencia de:
 - Lenguaje de programación
 - Aplicación
 - Plataforma
 - Dispositivo
 - Protocolos de transporte. Puede ser puesto en práctica sobre TCP/IP, HTTP, Bluetooth, HomePNA, y muchos otros protocolos

JXTA (pronunciado “yuxta”)

- Especificación de un conjunto de servicios y protocolos abiertos
 - permiten comunicación y colaboración P2P entre cualquier dispositivo conectado a la red independientemente de topología de red
 - crea una red virtual ad hoc sobre una red existente
- Componentes:
 - pares y grupos de pares
 - servicios
 - *pipes* (mecanismo para enviar mensajes)
 - mensajes (en implementaciones actuales: documentos XML)
 - anuncios (en implementaciones actuales: documentos XML)
- *JXTA development kit*
 - implementación de referencia en Java por Sun
 - JXTA es independiente de Java: cualquier lenguaje de programación, cualquier plataforma, cualquier fabricante
- Compara con PeerWare en <http://peerware.sourceforge.net/>

Arquitectura Software JXTA



Componentes JXTA

- **Peer:** Una entidad capaz de realizar alguna tarea útil y de comunicar los resultados a otra entidad en la red, ya sea directa o indirectamente.
- **Clasificación de peers:**
 - **Edge peers** : servir a un usuario, normalmente en una red privada
 - **Super peers**
 - **Rendezvous:** Coordina peers de una red JXTA y proporciona mecanismos para descubrir otros participantes, fuera de la red privada
 - **Relay:** permiten a los miembros de una red privada (detrás de un firewall participar en JXTA

Componentes JXTA

■ Grupos de participantes:

- Formado para servir un objetivo común.
 - Agrupación lógica de peers
 - Proporciona servicios a sus participantes miembros que no son accesibles a otros participantes en la red P2P.
- datos privados
- seguridad
- monitorización ente miembros
- Cada grupo debe tener al menos un rendezvous

Componentes JXTA

□ **Anuncios:**

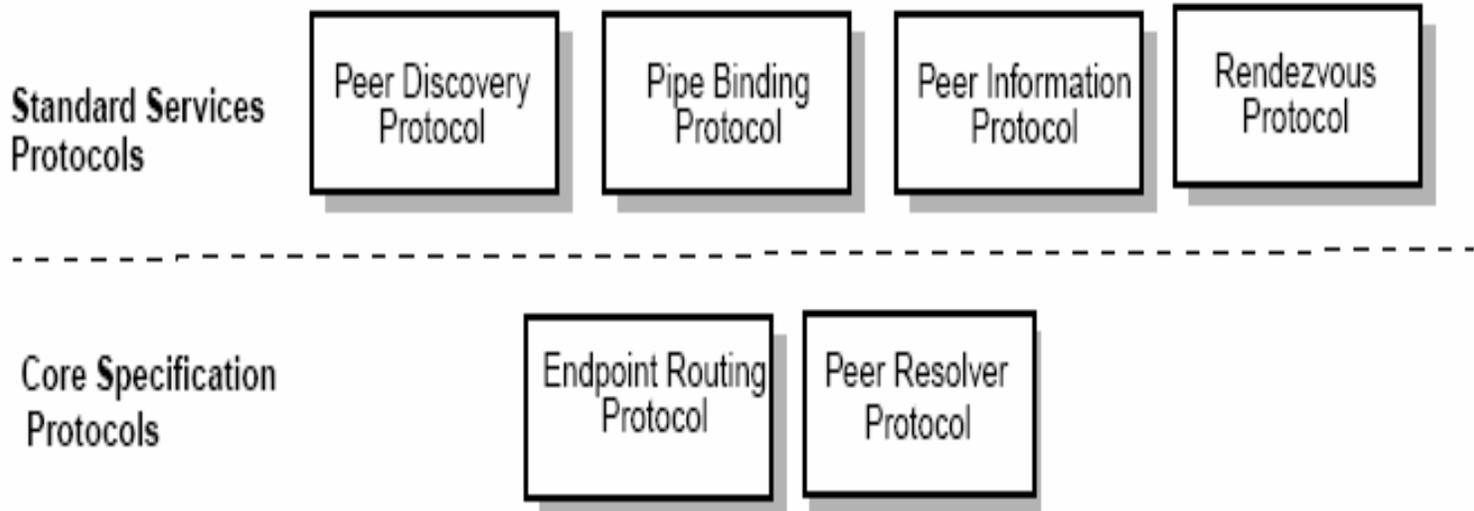
- Documentos XML que describen cualquier recurso en una red P2P (pares, grupos, conductos, servicios, etc.)

□ **Mensajes:**

- Contenedores de datos transmitidos por un conducto de un punto final a otro. Típicamente XML

Protocolos principales de JXTA (1/3)

- **Protocolo:** Mecanismo estructurado para el intercambio de información entre dos o más partes utilizando reglas previamente acordadas por todas las partes.



Protocolos principales de JXTA (2/3)

- Descubrimiento de pares (*peer discovery*)
 - utilizado para descubrir anuncios (*advertisements*) de otros pares del mismo grupo de pares
 - útil para descubrir pares, grupos de pares, *pipes* y servicios
- Resolución de pares (*peer resolution*)
 - utilizado para mandar una petición de búsqueda a otro par con el fin de localizar a pares, grupos de pares, un pipe o un servicio
- Suscripción/baja de pares (*peer membership*)
 - utilizado por pares para suscribirse o darse de baja de grupos

Protocolos principales de JXTA (3/3)

- Información de pares (*peer information*)
 - utilizado para conseguir información sobre el estado de otro par
- Ligadura de conductos (*pipe binding*)
 - utilizado por un par para ligarse al extremo de un conducto
- Encaminamiento (*endpoint routing*)
 - utilizado para pedir información sobre el camino hacia otro par

Servicios principales de JXTA (1/2)

- Conducto (*pipe*):
 - define el principal medio de comunicación entre pares
 - proporciona una abstracción de un conducto unidireccional y asíncrono para la transferencia de información
 - el extremo de un conducto puede o no estar conectado a un par y puede estar conectado a distintos pares en distintos momentos
 - dos tipos de conducto: 1-1 y 1-n (se está trabajando en n-m)
- Socio (*membership*):
 - determina los pares que pertenecen a un grupo de pares;
 - maneja la suscripción y baja de pares a los grupos
- Acceso (*access*):
 - seguridad
 - controla el acceso a los servicios y recursos dentro de un grupo de pares

Servicios principales de JXTA (2/2)

- Descubrimiento (*discovery*)
 - proporciona los medios para que los pares descubran a otros pares y a la existencia de grupos, *pipes* y servicios.
- Resolución (*resolver*)
 - permite que los pares hagan referencia a otros pares y a grupos, conductos (*pipes*), servicios etc. por medio de referencias / anuncios (*advertisements*)
 - liga referencias a implementaciones en tiempo real

El *shell* de comandos de JXTA

- Aplicación JXTA construida encima del *core*
 - un *shell* basada en Java está incluida en el *JXTA development kit*
- Especialmente útil para
 - entender los distintos componentes de la plataforma
 - prototipado
- Extensible
 - el usuario puede definir sus propios comandos *shell*
- Uso de variables de entorno como en los *shell* de Unix
- Comandos para
 - crear anuncios
 - crear pipes
 - crear y enviar mensajes
 - crear nuevos *shells*
 - ...

Mejoras JXTA 1.0 -> JXTA 2.0

- Soporta HTTP 1.1 (además de 1.0)
- Transporte TCP/IP bidireccional
- Retransmisión TCP/IP (además de HTTP)
- Separa funcionalidad de transmisión y transporte
- Descubrimiento dinámico de Rendezvous y Relays
- Mejora sistemas de almacenamiento y distribución de índices
- Soporta conexión asíncrona entre extremos de un pipe
- Nuevas características:
 - Mejora asignación de recursos para disminuir consumo de memoria
 - Soporta retransmisión de errores para mejorar fiabilidad de la red

Bibliografía

■ En la biblioteca:

- L/D 004.7 PEE 2001 **Peer-to-peer : harnessing the benefits of a disruptive technology** , Oram, Andrew

■ En la Web

□ En Inglés

- <http://zeropaid.com/>
- <http://www.slyck.com/>
- <http://www.openp2p.com/>
- <http://www.jxta.org>
- <http://www.spywareinfo.com/articles/p2p/>

□ En Español:

- <http://www.zonap2p.com/>
- <http://es.wikipedia.org/wiki/P2P>
- http://es.wikipedia.org/wiki/Historia_de_las_aplicaciones_P2P
- <http://www.vsantivirus.com/lista-p2p.htm>