

Sistemas Peer to Peer

Miguel Angel Astor Romero

SOA :: 2-2016

Agenda

- 1 Introducción
- 2 Clases de sistemas p2p
- 3 Problemas generales en los sistemas p2p
- 4 Ejemplos de Sistemas Peer-to-Peer
- 5 Conclusiones

Definición

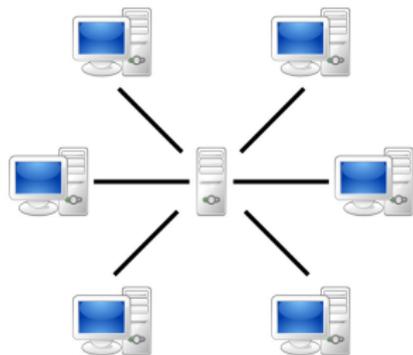
Sistema Peer-to-Peer

- Se define como sistema o red Peer-to-Peer (p2p) a un sistema distribuido donde los distintos nodos que componen al sistema pueden actuar simultáneamente como clientes y servidores para los demás nodos de la red.
- En otras palabras, en un sistema p2p todos los nodos son tratados como iguales en lo referente a la comunicación entre nodos (distribución horizontal), en contraposición con los sistemas cliente-servidor clásicos (distribución vertical).

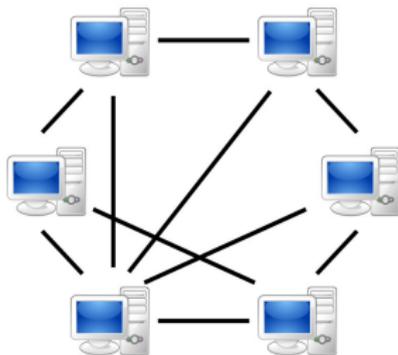
Distribución vertical vs. distribución horizontal

Figuras recuperadas de <https://en.wikipedia.org>.

Dist. vertical (cliente-servidor)



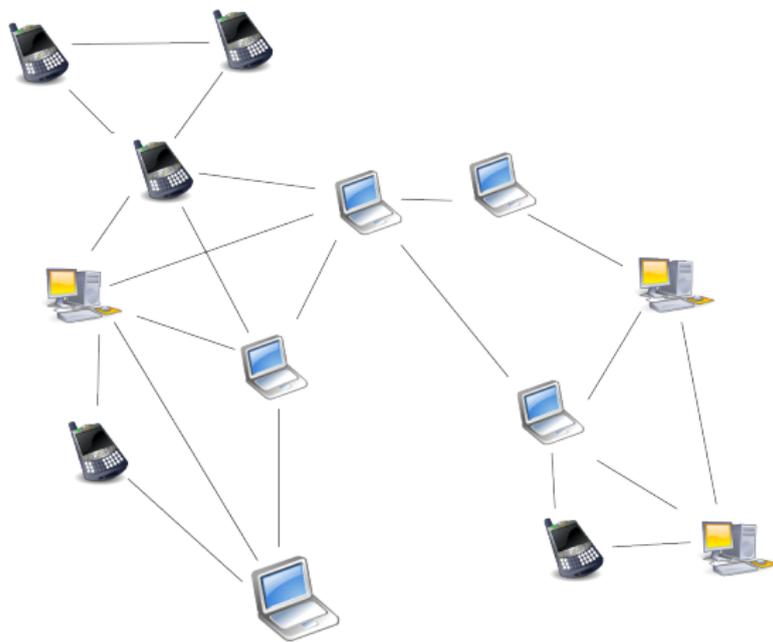
Distribución horizontal (p2p)



Red superpuesta (overlay network)

- En los sistemas p2p, los nodos que se comunican son procesos.
- Estos procesos forman canales virtuales de comunicación sobre la red UDP/TCP/IP subyacente.
- La red formada por estos canales de comunicación se conoce como red superpuesta o sobrepuesta.
- Los procesos solo se comunican con sus vecinos en la red superpuesta.
- Los enlaces de la red superpuesta no tienen porque corresponder con enlaces UDP/IP o TCP/IP reales.

Red superpuesta (overlay network)



Sistemas p2p según composición de la red

- Existen múltiples formas de componer la red superpuesta de un sistema p2p.
 - Sistemas p2p estructurados.
 - Sistemas p2p no estructurados.
 - Sistemas híbridos.

Sistemas p2p estructurados

- Los vecinos de cada nodo en la red se determinan de manera determinística.
- Insertar/eliminar nodos de la red es complicado.
- Las búsquedas de elementos o nodos en la red es muy eficiente ($O(\log(N))$ con N la cantidad de nodos en la red).

Sistema p2p estructurado

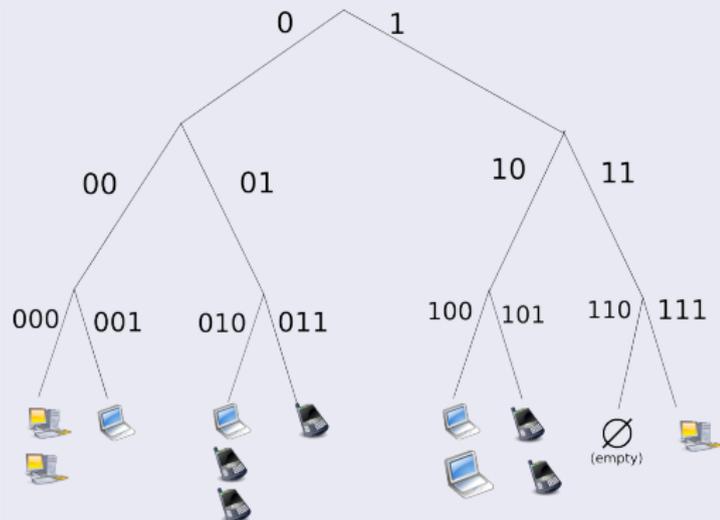


Imagen recuperada de <https://en.wikipedia.org>.

Sistemas p2p no estructurados

- Caso contrario.
- Los vecinos de cada nodo en la red se determinan al azar.
- Insertar/eliminar nodos de la red es sencillo.
- Las búsquedas son o muy lentas o muy ineficientes en cuanto al uso de la red.

Sistema p2p no estructurado

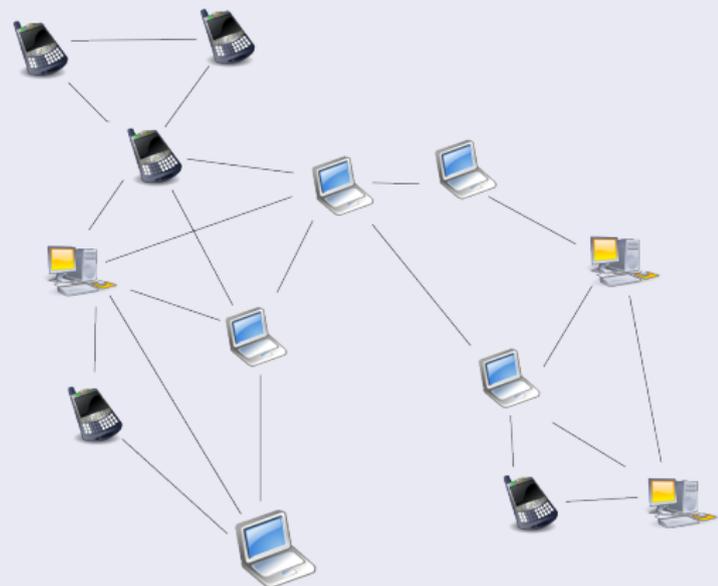


Imagen recuperada de <https://en.wikipedia.org>.

Arquitecturas híbridas

- Existen varias modalidades:
 - Uso de servidores centralizados para ciertas funciones del sistema.
 - Ciertos nodos son tratados de forma especial.

Estructura híbrida BitTorrent

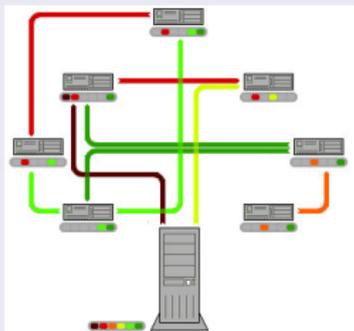


Imagen recuperada de <https://en.wikipedia.org>.

Red de superpares

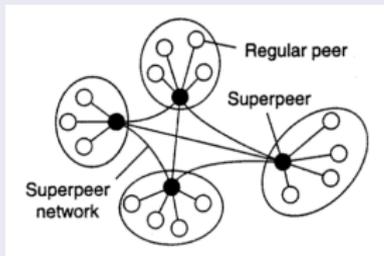


Figura recuperada de *(Tanenbaum, 2007)*

Problemas a resolver en sistemas p2p

- Para poder diseñar, implementar y utilizar sistemas p2p hay que considerar ciertos detalles:
 - Como determinar quienes son los vecinos de un nodo que se acaba de conectar a la red (problema de arranque).
 - Como realizar la inserción/eliminación de un nodo en la red (gestión de membresía).
 - Como buscar otros nodos o elementos de datos en la red (problema de búsqueda distribuida).

Arranque

- Cuando un nodo se conecta a la red este debe ser capaz de encontrar otros nodos para poder comunicarse (*bootstrap*).
- Una vez se completa el *bootstrap* del nodo, este puede descubrir nuevos pares comunicándose con sus vecinos (intercambio de pares).

¿Como puede hacer un nodo para encontrar posibles vecinos? Ideas.

Arranque

Soluciones al problema de arranque

- Cablear una lista de nodos conocidos en la aplicación.
- Utilizar servidores de nombres para solicitar pares iniciales:
 - *Trackers* en BitTorrent.
 - Semillas DNS en Bitcoin.
- Soluciones experimentales (ver lecturas recomendadas).

```

Terminal
[mikey]- $ nslookup seed.bitcoin.sipa.be
Server:      8.8.8.8
Address:    8.8.8.8#53

Non-authoritative answer:
Name:   seed.bitcoin.sipa.be
Address: 185.191.177.22
Name:   seed.bitcoin.sipa.be
Address: 68.65.120.49
Name:   seed.bitcoin.sipa.be
Address: 77.72.149.120
Name:   seed.bitcoin.sipa.be
Address: 138.68.79.47
Name:   seed.bitcoin.sipa.be
Address: 138.201.206.35
Name:   seed.bitcoin.sipa.be
Address: 198.37.118.11
Name:   seed.bitcoin.sipa.be
Address: 46.229.165.138
Name:   seed.bitcoin.sipa.be
Address: 185.77.128.69
Name:   seed.bitcoin.sipa.be
Address: 109.98.31.64
Name:   seed.bitcoin.sipa.be
Address: 23.243.251.82

```

Gestión de membresía

- En el caso de las redes no estructuradas la inserción/elminación de nodos en la red es sencilla:
 - Al insertar un nodo este adquiere una lista de vecinos con el mecanismo de arranque y se anuncia a estos.
 - Al recibir el anuncio, estos vecinos agregan la conexión con el nuevo nodo a la red superpuesta.
 - Si los nodos se comunican mediante tablas de rutas, estas se comparten con el nuevo nodo.
 - Si un nodo se va de la red, sus vecinos sencillamente eliminan sus enlaces con él de la red superpuesta y lo borran de sus tablas de rutas.
- En el caso de las redes estructuradas el proceso es más complicado.
- En estas redes es necesario que la inserción/eliminación del nodo sea lo menos disruptiva posible.
- En las redes como BitTorrent, los nodos que se insertan/eliminan sencillamente anuncian su intención de hacerlo a los servidores *tracker*.

Búsqueda

- Buscar en un sistema p2p se refiere tanto a conseguir un elemento de datos en posesión de un nodo, así como la información de conexión de este.
- Los mecanismos de búsqueda son diferentes si se trata de una red estructurada, no estructurada o híbrida.

¿Ideas?

Query flooding

La forma más sencilla de resolver el problema de las búsquedas en redes no estructuradas es sencillamente consultar a los vecinos recursivamente.

- *Query Flooding* (QF).
- El nodo iniciador consulta a un subconjunto de sus vecinos.
- Cada vecino consulta a un subconjunto de sus vecinos.
- Se limita una consulta a cierto número de saltos.
- Problemas:
 - No es escalable.
 - Ciclos.
 - Callejones sin salida.

Protocolo QF de Gnutella

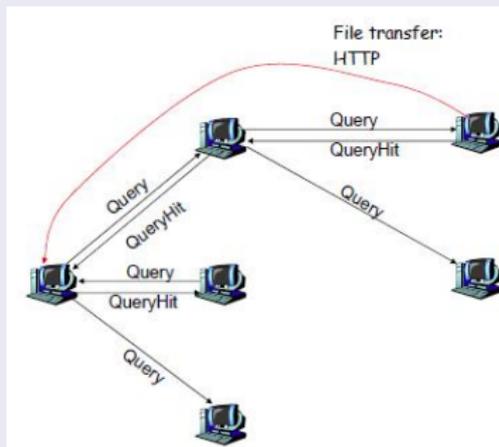


Imagen recuperada de <https://en.wikipedia.org>.

Recorridos aleatorios

Una mejora a los métodos *Query Flooding* es limitar la cantidad de vecinos que se consultan en cada salto.

- El nodo iniciador escoge a un vecino(s) al azar y le realiza la consulta.
- Se repite el proceso recursivamente hasta cierta profundidad de saltos.

Ventajas

Reduce considerablemente la cantidad de mensajes en la red.

Desventajas

El tiempo de respuesta es mayor y sujeto a una mayor varianza comparado con QF.

Query Routing Tables (QRT)

Otra idea es utilizar tablas de enrutamiento para almacenar que nodos son capaces de responder determinadas solicitudes. Este mecanismo fue inventado para el sistema de intercambio de archivos [Gnutella](#), basado en el [Query Routing Protocol](#).

- Conceptualmente similar a los protocolos vector-distancia.
- Los nodos se estratifican en “hojas” y “*ultrapeers*”. Las búsquedas son iniciadas por las hojas y ejecutadas por los *ultrapeers*.
- Los *ultrapeers* mantienen tablas cuyas filas contienen lo siguiente:
 - El *hash* de las palabras de la consulta.
 - ID del nodo que puede resolver esa consulta.
 - ID del nodo siguiente en dirección al nodo que puede resolver la consulta.
 - Distancia al nodo que puede resolver la consulta.
- Regularmente, los nodos intercambian sus tablas de rutas con sus vecinos.

Tablas Hash Distribuidas (DHT)

- Las DHT se utilizan en las redes p2p estructuradas.
- De hecho son las que establecen la estructura de la red.
- Se basan en la asociación de claves *hash* con elementos de búsqueda como en una tabla *hash* común.
- La idea es que al usar *hashes* para indexar nodos/elementos se
- Al diseñar una DHT es necesario plantearse las siguientes preguntas:
 - ¿Como se inserta un nodo a la gestión de la DHT?
 - ¿Que sucede si se elimina un nodo de la DHT?

Tablas Hash Distribuidas (DHT) - Implementaciones

Una forma común de implementar una DHT es con un anillo lógico de claves.

- Cada clave corresponde a un nodo o a un elemento de datos.
- Cada elemento de datos k se mapea al nodo de idk .
- Dos posibles implementaciones:
 - Doblemente enlazada.
 - Tablas finger.

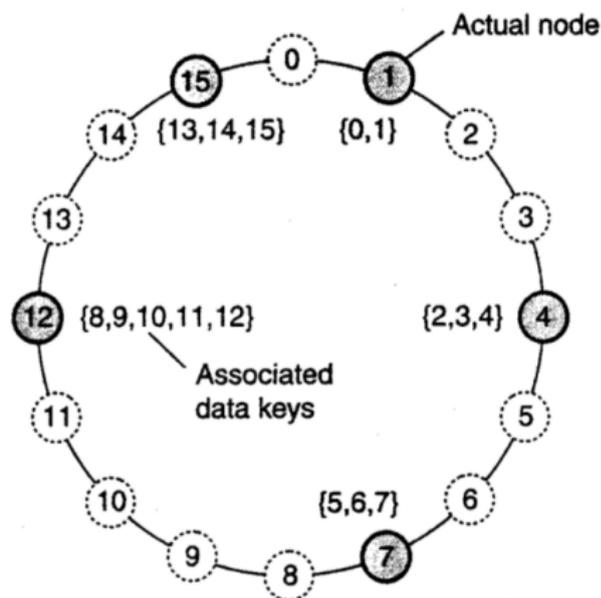


Figura recuperada de *(Tanenbaum, 2007)*

BitTorrent

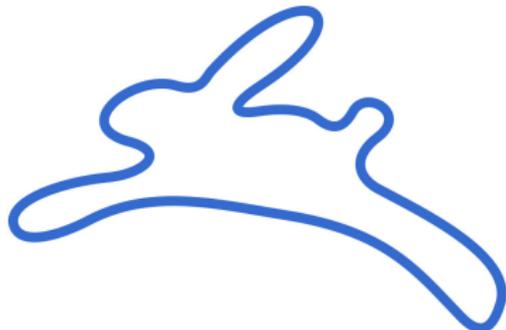
BitTorrent es una red p2p para intercambio de archivos.

- Arquitectura híbrida estructurada.
- Se usan servidores *tracker* para el *bootstrapping*.
- Se usan DHT para intercambio de pares.



Freenet

- Almacén de datos distribuido y framework (Java) para la construcción de servicios p2p anónimos y resistentes a la censura.
 - Mensajería.
 - Compartición de archivos.
 - *Freesites*.
- Basado en búsquedas de claves con enrutamiento (similar a DHT).
- Funciona en modo p2p (*opennet*) o Friend-to-Friend (*darknet*) según un esquema *web-of-trust*.





- Motor de búsqueda distribuido.
- Compartición de índices p2p.
- Basado en DHT.
- Cada nodo funciona como Web crawler.
- Escrito en Java.

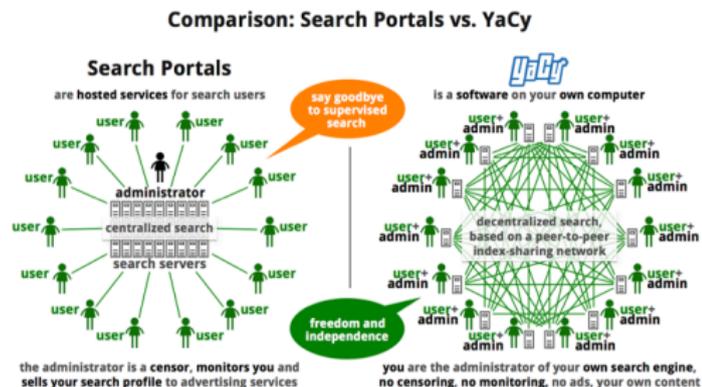


Figura recuperada de [*\[\[http://http://yacy.net\]\]*](http://http://yacy.net)

Otras redes y herramientas p2p

- **Kademlia**: Protocolo para implementar DHT's.
- **ZeroNet**: Web anónima distribuida, resistente a la censura.
- **Direct Connect**: Protocolo para mensajería e intercambio de archivos.
- **eDonkey**: Protocolo p2p libre para intercambio de archivos.
- **Gnutella**: Primera red p2p descentralizada y no estructurada.
- **I2P**: Red p2p de enrutamiento para navegación Web anónima.
- **Tradepal**: Mercado descentralizado en línea.
- **Bitcoin**: Sistema monetario descentralizado.
- **GNUnet**: Framework para la construcción de sistemas p2p en C.

Bibliografía

- Tanenbaum, A. y M. Van Steen, [Sistemas Distribuidos: Principios y Paradigmas](#), 2ª Edición Pearson, 2007.
- Korzun, D. y A. Gurtov, [Structured Peer-to-Peer Systems](#), 1ª edición, Springer, 2013.
- Gribble, S. et al., [Scalable, Distributed Data Structures for Internet Service Construction](#), Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4, pp. 22, USENIX Association, octubre, 2000.
- Lv, Q. et al., [Search and Replication in Unstructured Peer-to-Peer Networks](#), Proceedings of the 16th international conference on Supercomputing, pp. 84-95, ACM, junio, 2002.
- Clarke, I. et al., [Freenet: A Distributed Anonymous Information Storage and Retrieval System](#), Designing Privacy Enhancing Technologies, pp. 46-66, Springer, 2001.

Lecturas recomendadas

- GauthierDickey, C. y C. Grothoff, **Bootstrapping of Peer-to-Peer Networks**, International Symposium on Applications and the Internet, pp. 205-208, IEEE, 2008.
- Balakrishnan, H. et al., **Looking up Data in P2P Systems**, Communications of the ACM, Vol 46(2), pp. 43-48, ACM, 2003.
- Clarke, I. y C. Mellish, **A Distributed Decentralised Information Storage and Retrieval System**, Disertación doctoral, University of Edinburgh, 1999.
- Ratnasamy, S. et al. **A Scalable Content-Addressable Network**, Transactions of the ACM, Vol 31(4), pp. 161-172, ACM, 2001.
- Wiley, B., **Distributed Hash Tables**, disponible en <http://www.linuxjournal.com>, 2003.

¿Preguntas?

