

# Redes entre pares (peer-to-peer)

---

# Concepto

---

- Se habla de redes entre pares cuando el servicio se obtiene sin servidores o control central.
- Se usan para que unos nodos (en Internet) compartan parte de sus recursos con otros nodos.

# Sistemas cliente/servidor

---

- Sistema 'tradicional'.
- Un servidor único, bien conocido y fiable (supuestamente).
- Sencillo de implementar.
- Sencillo de administrar.
- Modelo con éxito (www, dns, etc).

# Sistemas cliente/servidor (II)

---

- Tienen algunos problemas:
  - Escalabilidad.
  - Punto único de fallo (atraen atacantes).
  - Requiere administración (replicación por seguridad, equilibrio de carga, etc).
  - Se desaprovechan recursos.

# Sistemas entre pares

---

- Nuevo paradigma.
- Todos los nodos tienen la misma funcionalidad, rol o misión.
- No hay servidor y clientes, todos los nodos son servidores Y clientes.
- Todos los nodos ofrecen y consumen ***recursos.***

# Recursos

---

- Capacidad de almacenamiento e información. Ej.: eDonkey, eMule.
- Capacidad de comunicación. Ej.: Redes superpuestas.
- Capacidad de cómputo. Ej.: SETI@home

# Características de Sists. P2P

---

- Los nodos actúan como servidores y enrutadores.
- Los nodos son autónomos, y no existe autoridad central.
- Intercambio y comunicación directa entre los nodos.
- Los nodos entran y salen de la red con frecuencia.

# Ventajas de Sists. P2P

---

- Uso más eficiente de recursos.
- Escalabilidad. A más usuarios, también más recursos.
- Fiabilidad. No hay punto único de fallo. Replicación de la información.
- No necesitan administración.

# Problemática

---

- Uso/acceso/descarga del recurso.
- Localización del recurso.

Ejemplo: Gnutella

- Se usa el sistema para localizar un par que tiene un fichero.
- Se descarga el mismo del par.

# Acceso al recurso

---

- Descarga/uso del par que tiene el recurso directamente.
- Colaboración entre pares con réplicas (totales o parciales) para descargas.
- Transferencia de programas y datos. Ejecución controlada.

# Localización

---

- Servicio centralizado:
  - Un solo servidor (Napster).
  - Árboles de servidores (fasttrack).
- Servicio no centralizado:
  - Red estructurada, DHT (Chord, Pastry).
  - Red no estructurada (Gnutella, Freenet).

# Localización Centralizada

---

- Repite las mismas características de los sistemas cliente/servidor.
- Mismas ventajas e inconvenientes.

# Localización No Centralizada

---

- Los nodos forman conexiones virtuales para construir una red superpuesta (*overlay*).
- Las búsquedas son enrutadas a través de la red overlay. Un nodo sólo manda búsquedas a sus vecinos.
- Sistemas estructurados/ no estructurados.

# Redes Estructuradas / DHT

---

- Cada recurso es 'asignado' a un nodo. Este debe de saber dónde está localizado el recurso (aunque no lo gestione/posea).
- Hay un control estricto de las conexiones de la red superpuesta.
- Se basan en el uso de Tablas Hash Distribuidas (DHTs).

# Redes No Estructuradas

---

- Un recurso puede encontrarse en cualquier nodo.
- No hay, a priori, restricciones en quien puede estar conectado a quien.
- Sin garantías de encontrar el recurso.

# Inundación

---

- Cada nodo reenvía cada mensaje de búsqueda a todos sus vecinos.
- NO escalable.
- Usado por Gnutella.

# Super-Peers

---

- Ciertos nodos especiales, de gran capacidad, gestionan las búsquedas. Son los super-peers.
- El resto de nodos se conectan a ellos.
- Cada super-peer conoce los recursos de todos los pares conectados a él.

# Super-Peers (II)

---

- Cada búsqueda iniciada por un nodo es enviada a su super-peer.
- Si el super-peer no conoce la respuesta, puede reenviar la búsqueda a otros super-peers.
- Problema: dependencia de super-pares.
- eDonkey, Kazaa (FastTrack prot).

# Caminos aleatorios

---

- Cada mensaje de búsqueda es reenviado a un único vecino en la red overlay.
- Mecanismo de búsqueda 'ciego'.
- Consume pocos recursos de red.
- Eficiencia?
  - Posibles falsos negativos.
  - Tiempo para encontrar un recurso?.

# Caminos aleatorios (II)

---

- Se sabe que la topología de la red superpuesta influye en la eficiencia.
- Se investiga en sistemas con:
  - Topologías adaptativas.
  - *One-hop replication, cada nodo conoce los recursos de sus vecinos.*

# GIA

---

- Los nodos intentan estar lo más conectados posible.
- Se usan mecanismos de negociación para evitar saturar nodos.
- Las búsquedas se dirigen a nodos bien conectados.

# DANTE

---

- Busca topologías lo más centralizadas posible.
- Se basa en un heurístico sencillo, que indica a qué otros pares conectarse.
- No necesita mecanismos de negociación para evitar saturación.