Sistemas Distribuidos
Módulo 4

Sistemas Peer-to-Peer

Agenda

1. Característica
2. Evolución
3. Aplicaciones
4. Primera, Segunda y Tercera Generación
5. Métodos de Búsqueda. DHT

Agenda

1. Característica
2. Evolución
3. Aplicaciones
4. Primera, Segunda y Tercera Generación
5. Métodos de Búsqueda. DHT

Peer-to-Peer (P2P)

«Peer-to-peer es un Sistema auto-organizado de entidades iguales, autónomas (peers), cuyo objetivo es el uso compartido de recursos distribuidos en un ambiente de red evitando servicios centralizados» Oram y otros, 2001.

Peer-to-Peer (P2P)

P2P representa un paradigma para la construcción de sistemas distribuidos y aplicaciones en los cuales datos y recursos computacionales son provistos por varios huéspedes en Internet, todos los cuales participan en la provisión de un servicio uniforme.

Un problema clave es la **UBICACIÓN** de los objetos de datos en muchos huéspedes y el subsecuente **ACCESO** a los mismos de manera de balancear la carga y asegurar la disponibilidad sin agregar sobrecarga.

P2P: Características

- Cada usuario **CONTRIBUYE** con recursos al sistema. No necesariamente con el mismo recurso.
- Todos los nodos en un sistema P2P tienen la **MISMA CAPACIDAD** funcional y reponsabilidades.
- Su correcta administración no depende de la existencia de un sistema administrativo central.
- Pueden ser diseñados para ofrecer un limitado grado de anonimato a provedores y usuarios de recursos.
- Un aspecto clave para su eficiente operación es la elección de un algoritmo de ubicación de los datos a lo largo de muchos huéspedes y su subsecuente acceso.

Agenda

1. Característica
2. Evolución
3. Aplicaciones
4. Primera, Segunda y Tercera Generación
5. Métodos de Búsqueda. DHT

Desde ARPANET hasta Peer-to-Peer

- Fines de los 1960: Establecimiento de ARPANET
  - Objetivo: compartir recursos de computación y documentos entre centros de investigación en EEUU.
  - Aplicaciones: FTP y TelNet → modo cliente-servidor, con búsqueda y almacenamiento centralizado.
  - Un comité directivo central organiza la red.

- 1979: Desarrollo del protocolo de UseNet
  - Aplicaciones de newsgroup para organizar contenido.
  - Autoorganización para agregar o remover servidores de newsgroup.
  - La aplicación es aún cliente- servidor.

Desde ARPANET hasta Peer-to-Peer

- ~1990 corrida del público en general hacia Internet
  - Las aplicaciones siguen el enfoque cliente-servidor: WWW, email, streaming.
  - Basados en conexiones por modem vía SLIP y PPP protocol.
  - Modelo directo para administrar y controlar la distribución de contenido.
  - La seguridad resulta en un particionamiento por firewalls en Internet.

La Historia de Napster

- Mayo 1999: Disrupción de la comunidad Internet Primera Generación de P2P
  - El usuario consume contenido y ofrece contenido a otros participantes.
  - Los usuarios establecen una red virtual, enteramente independiente de la red física y autoridades administrativas o restricciones.
  - Bases: conexiones UDP y TCP entre los peers.
- Diciembre 1999: denuncia contra Napster Inc.
  - Blanco: el servidor de búsqueda centralizada de Napster.
- Febrero 2001: 2.79 miles de millones de archivos intercambiados vía la red Napster por mes
- Julio 2001: Napster Inc. es condenado
  - Napster tiene que parar la operación del servidor Napster.

Gnutella y sus Parientes

- Marzo 2000: Nullsoft lanza Gnutella como un proyecto de fuente abierta
  - Adicionalmente a la funcionalidad del servent, los peers hacen tareas de ruteo.
  - Completamente decentralizados.
- Octubre 2000: introducción de capas jerárquicas de ruteo.
  - Segunda Generación de P2P
  - Gnutella: reflector/concepto de Superpeer
  - Incrementa la escalabilidad significativamente.
- Variedad de protocolos decentralizados P2P:
  - Audiogalaxy
  - FastTrack/KaZaA
  - iMesh
  - Freenet

Gnutella y sus Parientes. La historia continúa

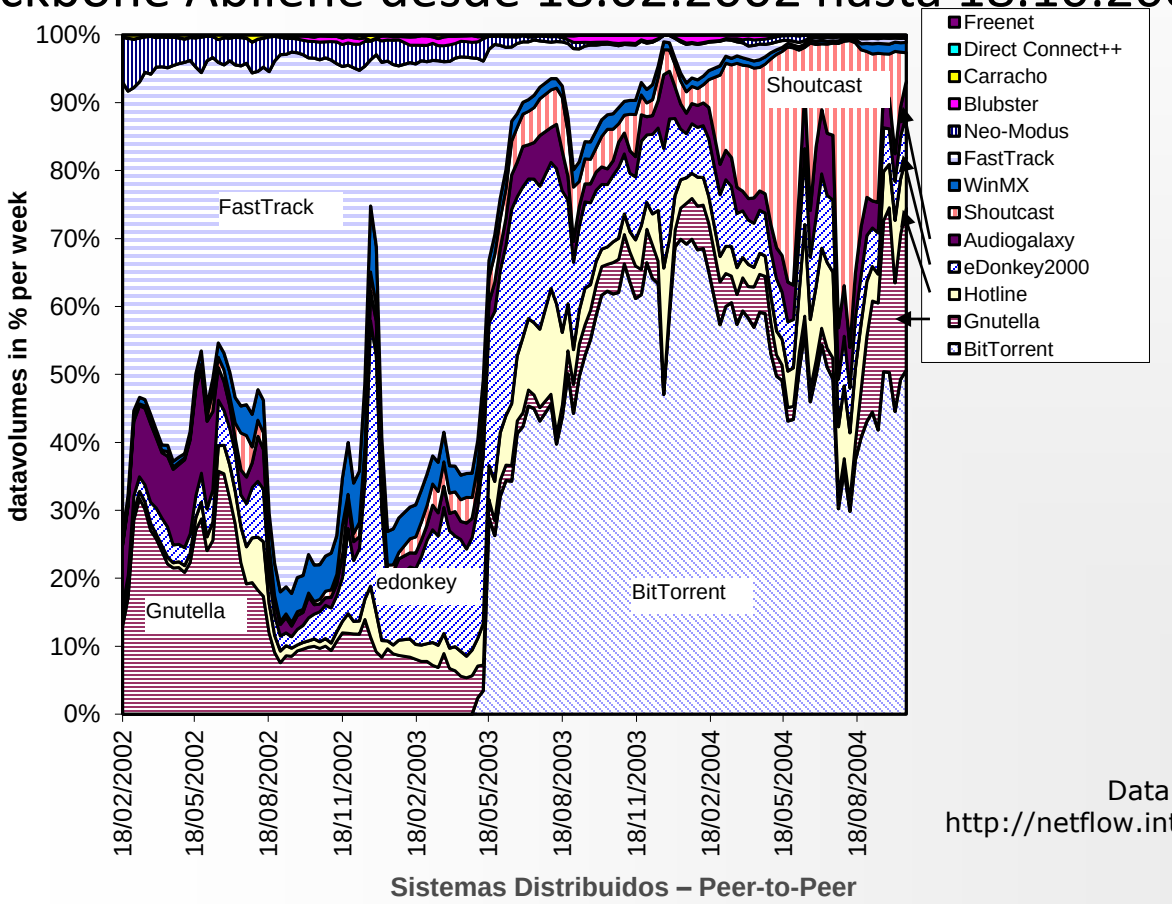
- Agosto 2001
  - Los usuarios se adaptan muy rápido a la caída de Napster
  - Casi 3.050 millones de archivos intercambiados por mes vía red Gnutella
- Año 2001: Tercera Generación de P2P se inicia
  - La primera investigación que comienza la tercer generación son las redes estructuradas P2P.
  - Características básicas: Uso de un algoritmo proactivo de ruteo basado en tablas Distributed Hash (DHTs).
- Agosto 2002
  - El intercambio de datos en KaZaA cae causado por el gran número de archivos defectuosos (razón: claves hash débiles para identificar archivos).
  - Edonkey y Gnutella ganan popularidad.

Gnutella y sus Parientes. La historia continúa

- Mayo 2003
  - Aparece Bittorrent
  - Se observa de pronto mayor tráfico. Razón: su popularidad, pero también que los datos de usuarios es intercambiado vía canales señalados en contraste con Gnutella, edonkey,…
- Mediados de 2003
  - Mas allá de intercambios de contenidos, se desarrollan nuevos conceptos para usar P2P en otras aplicaciones
  - Se desarrolla Skype sobre P2P
- …
  - Se trata de incrementar la confiabilidad de las búsquedas P2P, para usar P2P en redes móviles.
  - Ebay compra a mediados de 2005 Skype para usar el paradigma de comunicación entre compradores y vendedores

Desarrollo de Aplicaciones P2P

- Porciones de tráfico de las diferentes aplicaciones y protocolos P2P en las métricas de tráfico obtenidas por semana en el backbone Abilene desde 18.02.2002 hasta 18.10.2004



Data source:
http://netflow.internet2.edu/weekly/

Tráfico en las redes

Traffic Growth	1 Feb – 1 Mar Growth	1 Mar – 1 Apr Growth	1 Apr – 19 Apr Growth	Total 1 Feb – 19 Apr
Upstream	+2.07%	+123.18%	-2.95%	+121%
Downstream	-2.39%	+11.72%	+13.67%	+23%
Total	-1.74%	+28.69%	+9.28%	+38%

Data source:
Reporte especial por el Covid-19 realizado por Sandvine

Tráfico en las redes

- Buscar reportes sobre el tráfico de las redes en https://www.sandvine.com
- Reporte especial por el covid-19 en https://www.sandvine.com/covid-internet-spotlight-report?hsCtaTracking=69c3275d-0a47-4def-b46d-506266477a50%7Cac52173f-34c1-42df-8469-a091e7219e7a

Agenda

1. Característica
2. Evolución
3. Aplicaciones
4. Primera, Segunda y Tercera Generación
5. Métodos de Búsqueda. DHT

Las Aplicaciones P2P puedes ser clasificadas por los recursos compartidos

Clasificación Convencional Clasificación de P2P

- **Archivos Compartidos (Napster, Gnutella, Freenet)**
- **Grid Computing (SETI@home)**
- **Mensajería Instantánea (ICQ, AIM)**
- **Colaboración (Espacio de trabajo Groove)**

Clasificación por medio de recursos compartidos

- **Información**
- **Archivos**
- **Ancho de Banda**
- **Espacio de almacenaje**
- **Ciclos de Procesador**

- No hay una clara distinción
- Algunos casos tienden a confundir

Presencia de Información

- Importante rol en la autoorganización de redes P2P y en escenarios relacionados con computadoras omnipresentes y disponibilidad de la información (ubiquitous computing).
- Provee información acerca de cuales peers y que recursos están disponibles en la red.
- Ejemplo: *Sistemas de Mensajería Instantánea*
  - Aplicaciones P2P que usan esencialmente la presencia de información.
  - Peers pasan información vía la red, si están o no disponibles para comunicación.

Manejo de Documentos

- Usualmente organizados en forma centralizada
- Pero
  - Grandes porciones de documentos creados en una compañía son distribuidos entre PCs sin un repositorio central que tenga algún conocimiento de su existencia.
- Solución
  - Las redes P2P crean un repositorio conectado de los datos locales de los *peers* individuales.
  - Indexación y categorización de datos por cada *peer* sobre la base de un criterio seleccionado individualmente.
  - Agregado de información autoorganizada desde áreas del conocimiento.

Archivos

- *Archivos compartidos:*
  - Almacena contenidos en nodos individuales en vez de un lugar central.
  - Peers que bajan archivos los ponen disponibles para otros peers.
  - El 70% del tráfico de Internet puede ser atribuido a intercambio de archivos.
  - Es la aplicación más extendida de los sistemas P2P.

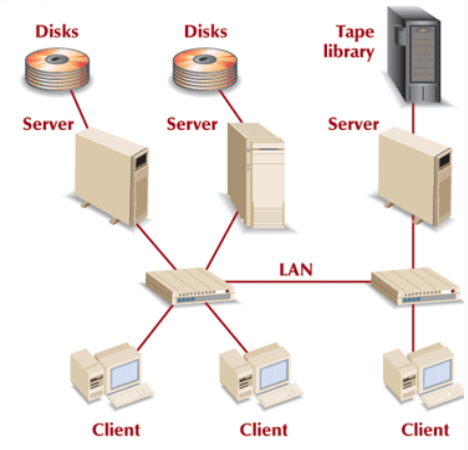
Búsqueda

- *Problema de Búsqueda:*
  - Localizar recursos es, en general, un problema central de las redes P2P y compartir archivos en particular.
  - Las redes P2P proveen diferentes métodos para almacenaje, búsqueda y recuperación de los archivos:
    - Modelo de Directorio Centralizado.
    - Modelo de requerimieno por “inundación” .
    - Modelo de ruteo de Documentos.

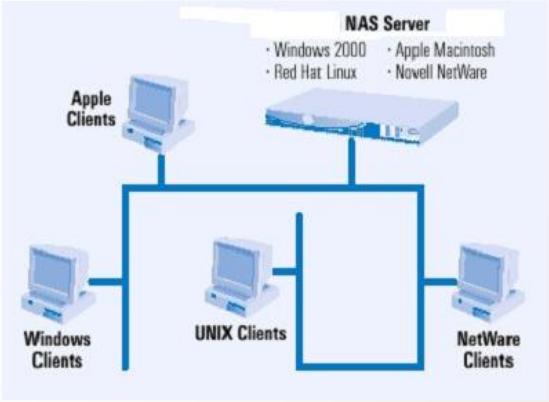
Espacio de Almacenamiento

- Conceptos de Diseños Centralizados Usados para Datos en una Compañía **DAS, NAS, SAN**:

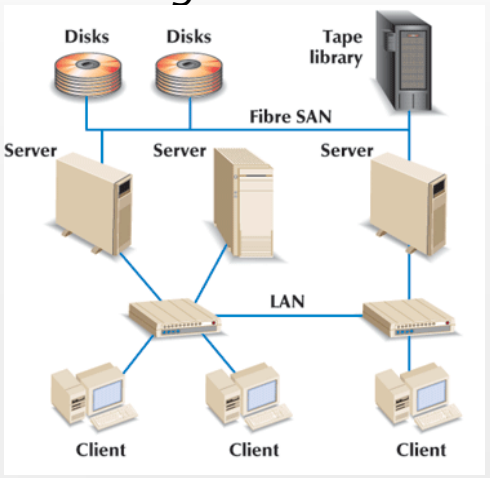
- *Direct Attached Storage (DAS)*



- *Network Attached Storage (NAS)*



- *Storage Area*



Espacio de Almacenamiento

- Una *Red de Almacenaje P2P* es un cluster de computadoras, formado sobre la base de redes existentes, que comparten todo el almacenaje disponible en la red.
  - Ejemplos: PAST, Pasta, OceanStore.

- Organización:
  - Cada *peer* recibe un par de claves pública/privada.
  - Cada peer debe tener disponible algo de su propio almacenaje.
  - A cada *peer* le es asignado un volumen máximo de datos que puede ser agregado a la red.
  - A un archivo le es asignado un número de identificación no ambiguo.
  - Almacenar y buscar un archivo en la red se hace de la manera descripta en el modelo de ruteo de documentos.

Ciclos de Procesador

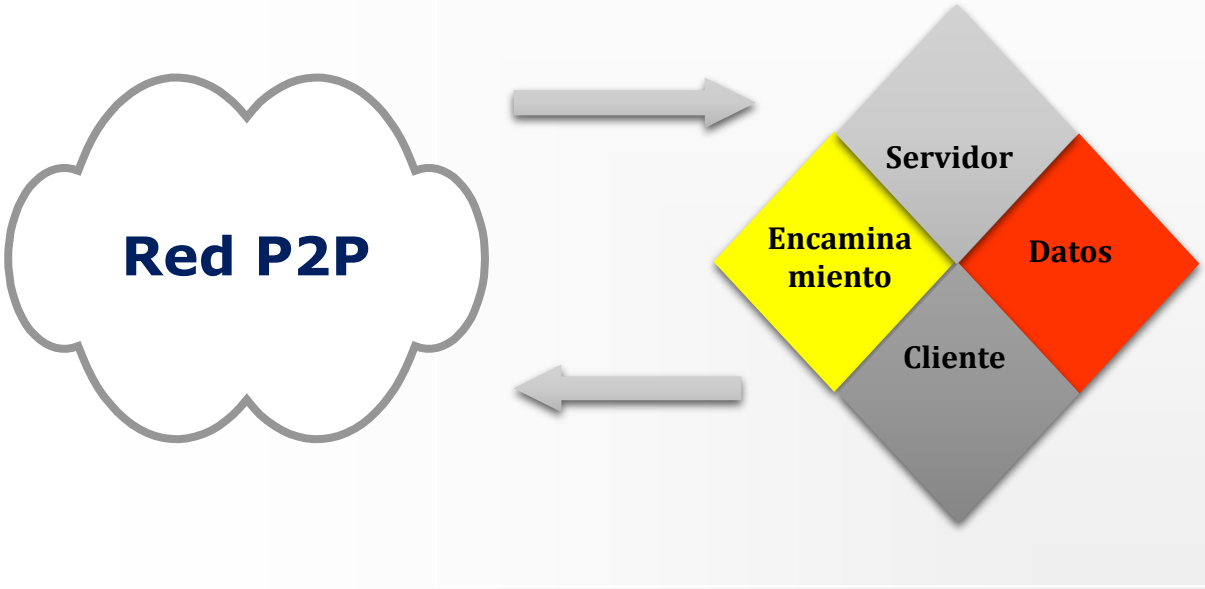
- Incremento de Requerimientos para Computación de Alto Rendimiento
  - i.e. campo de la bio-informática, logística o sector financiero.
- Poder de computación de entidades en red con poco uso.
- Uso de aplicaciones P2P para uso de ciclos de procesador:
  - Cluster de computadoras conectadas en red que como una única computadora es transparente y todos los nodos en red son combinados en una computadora lógica simple. Lograr poder de cómputo.
  - *"Grid Computing"*
- Ejemplos:
  - Ejemplo popular: *SETI@home*
    - Calcula durante ciclos ociosos del procesador de los peers participantes.
  - Visón advanzada de grid computing: *Globus Toolkit*
    - Middleware standard para aplicaciones grid.

Nota:
El núcleo de SETI@home es una clásica aplicación Cliente-Servidor.

Agenda

1. Característica
2. Evolución
3. Aplicaciones
4. Primera, Segunda y Tercera Generación
5. Métodos de Búsqueda. DHT

Servents

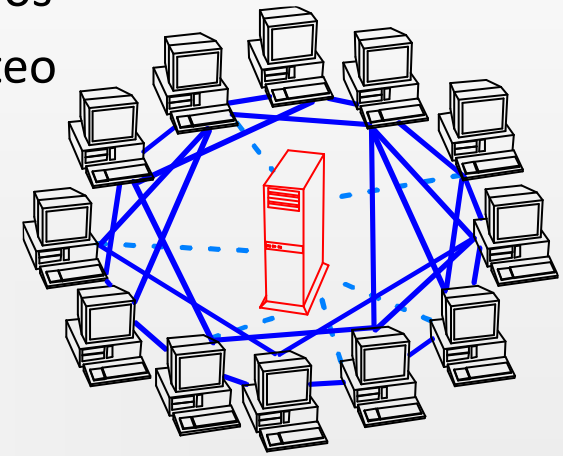


Modelo de Directorio Centralizado

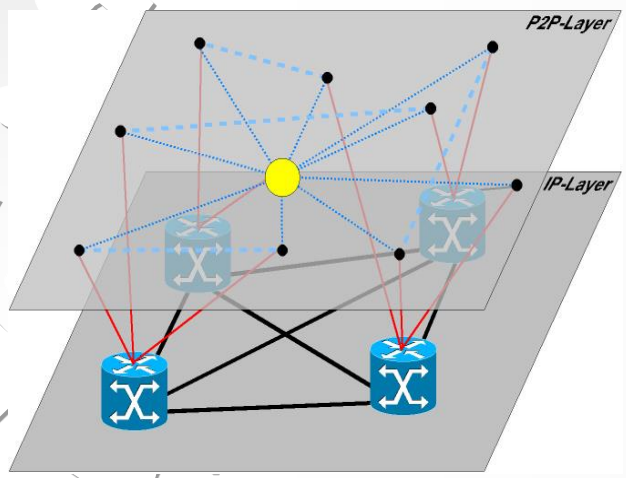
- ***Modelo de Directorio Centralizado:***
  - Ejemplo perfecto de sistema P2P híbrido
    - El servicio de indexado es provisto centralmente por una entidad de coordinación.
    - Un requerimiento de búsqueda es atendido por la entidad coordinadora que presenta una lista de peers que tienen los archivos requeridos.
    - Luego el peer obtiene los archivos respectivos directamente de los otros peers que los ofrecen.
- Características:
  - La búsqueda de documentos existentes puede ser garantizada.
  - El servicio de indexado es "**Un Único Punto de Falla".**
- Ejemplo: *"Primer" Napster*

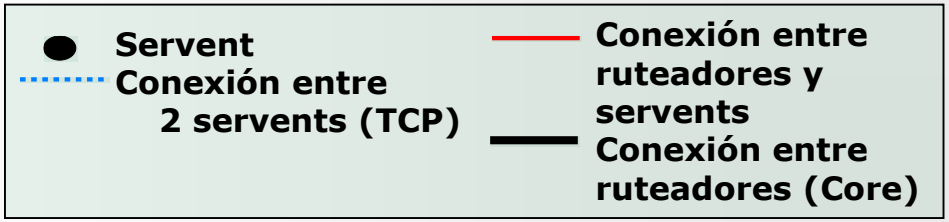
Definición de P2P centralizada

- Todos los peers están conectados a una entidad central
- Los Peers establecen conexiones entre ellos para intercambiar datos (e.g. mp3)
- La entidad central es necesaria para proveer el servicio.
- La entidad central es una clase DB índice/grupos
- La entidad central es la tabla de búsqueda/ruteo

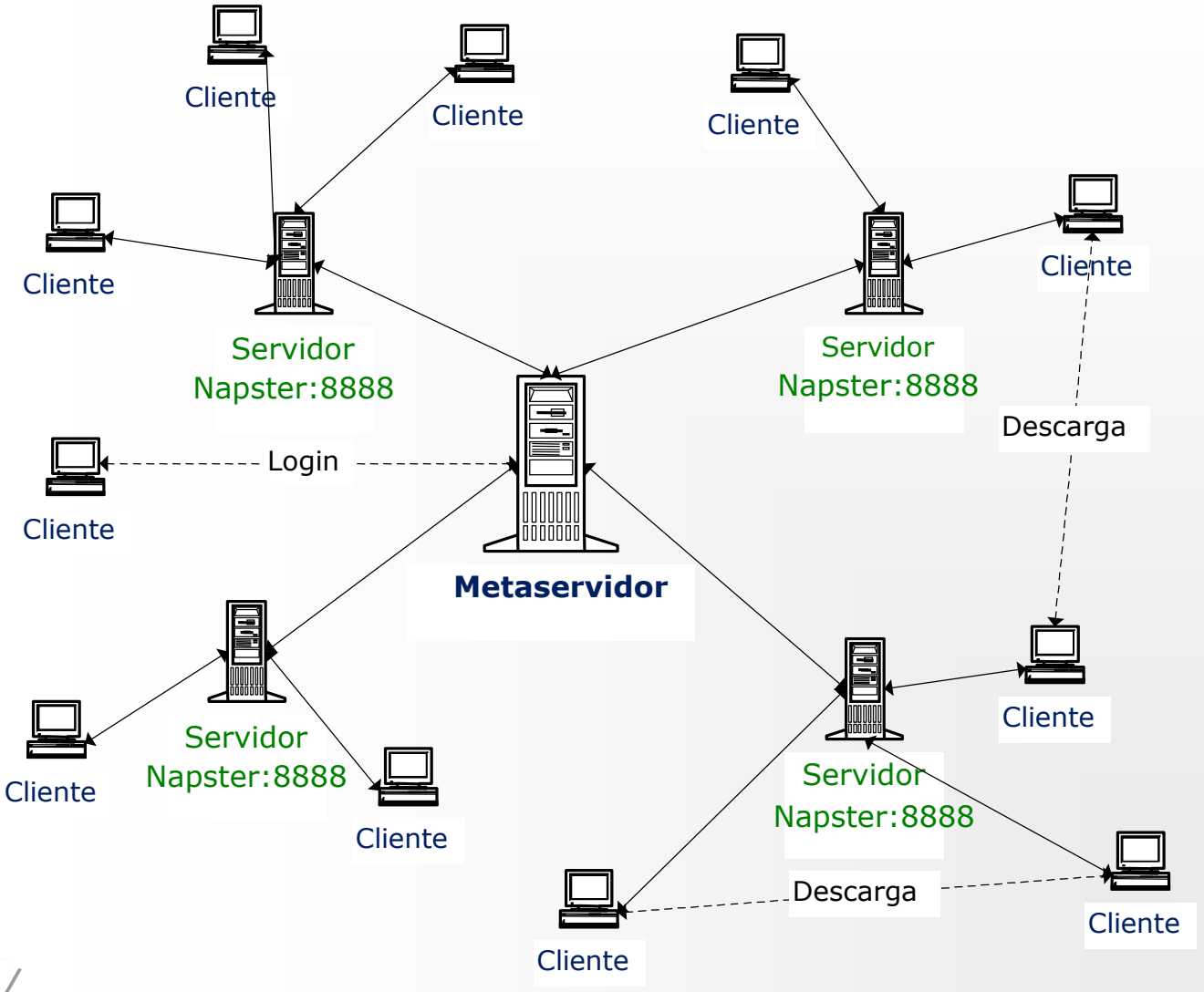


Topología

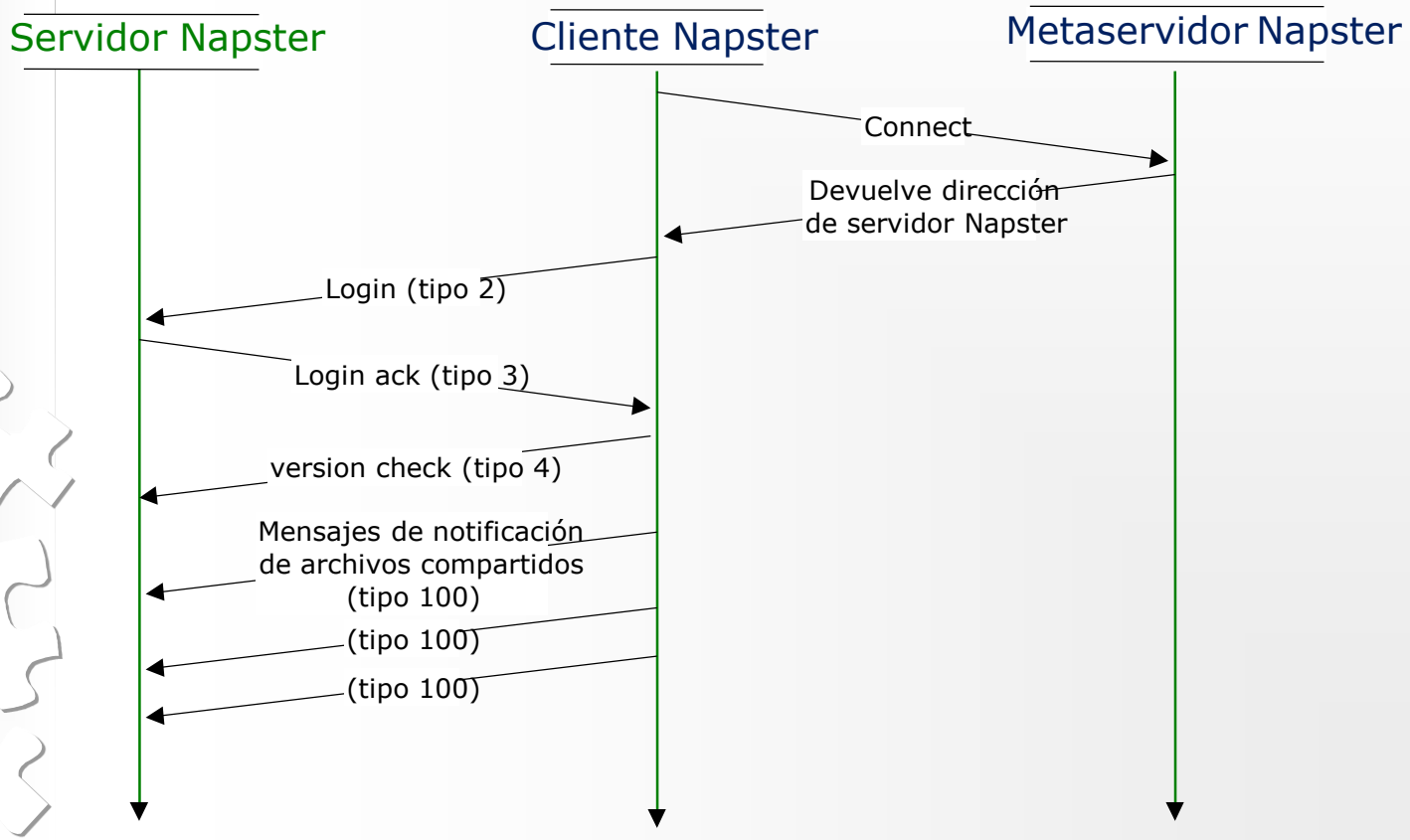




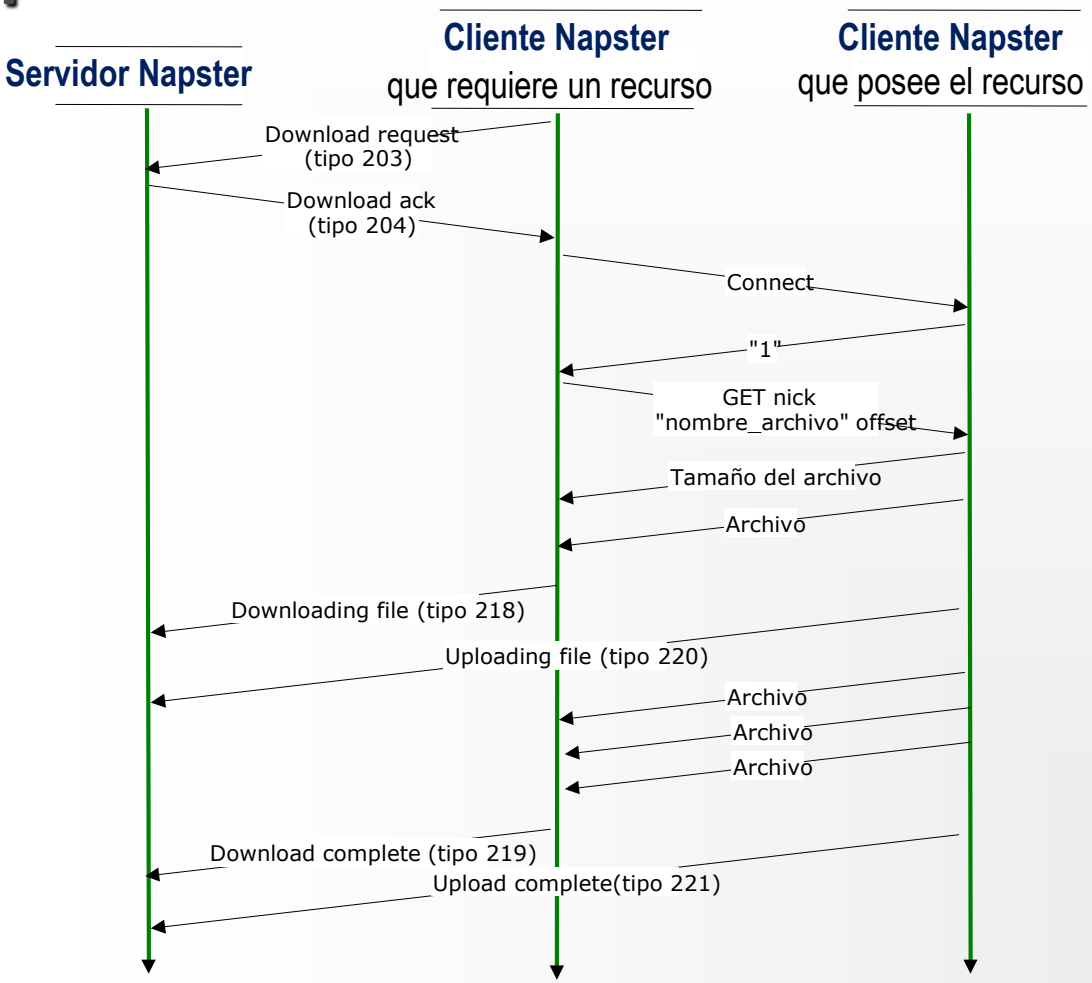
Napster



Napster



Napster



Modelo de Requerimiento por “Inundación”

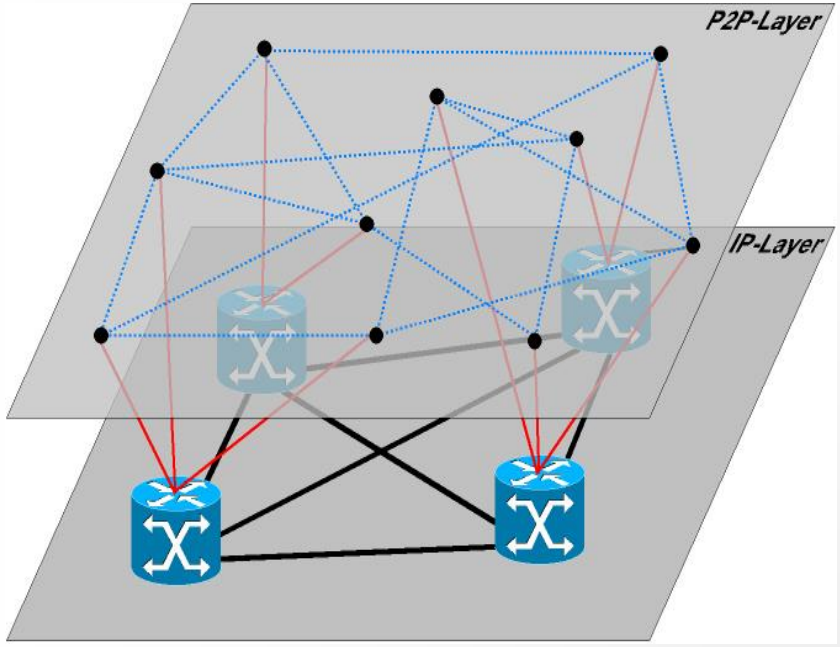
- ***Modelo de Requerimiento por “Inundación”:***
  - **Sistema P2P atómico**
    - Sin autoridad central de coordinación.
    - El requerimiento de búsqueda pasa por un determinado número de peers.
    - Si no pueden responder el requerimiento, lo pasan sobre otros nodos hasta cierto nivel de profundidad (ttl=time-to-live).
    - Cuando el archivo requerido ha sido localizado, los resultados positivos de la búsqueda son enviados a la entidad que los requirió.
    - El peer puede ahora bajar el archivo deseado directamente de las entidades que lo ofrecen.

Modelo de Requerimiento por “Inundación”

- Características:
  - La búsqueda de documentos existentes no puede ser garantizada.
  - El sistema no escala bien.

- Ejemplo: *Gnutella*
  - Extensión: *FastTrack*
    - “Super Peers” como proxies.
    - El sistema P2P no es más atómico.

Topología de P2P puro



- Servent
- Conexión entre 2 servents (TCP)
- Conexión entre ruteadores y servent
- Conexión entre ruteadores (Core)

P2P Puro - Gnutella

- Constituye una red solapada virtual con mecanismos propios de ruteo
- No hay coordinación central de actividades en la red.
- El software de soporte son “servents”.
- Usa IP como red subyacente, mientras que la comunicación entre servents está especificada en forma de protocolo a nivel de aplicación.

P2P Puro – Gnutella: Protocolo

Luego de unirse a Gnutella (conectándose a los nodos encontrados en la base de datos gnutellahost.com), el nodo envía un Ping al nodo al cual está conectado. Los nodos responden Pong identificándose y propagando el Ping a sus vecinos.

La localización de un archivo se hace por una búsqueda no determinística dado que los nodos no pueden adivinar donde están los archivos.

Gnutella usa broadcast para distribuir Ping y Query.

P2P Puro – Gnutella: Protocolo

Cada nodo los envía a sus vecinos y las respuestas hacen el camino inverso.

Para limitar los mensajes los mismos tienen un TTL (time to live), en cada hop este TTL es decrementado y cuando llega a 0 el mensaje se “cae”

Se eliminan lazos de mensajes por medio de un identificador.

Una vez que el nodo recibe un QueryHints inicia con esos datos una conexión directa entre dos nodos.

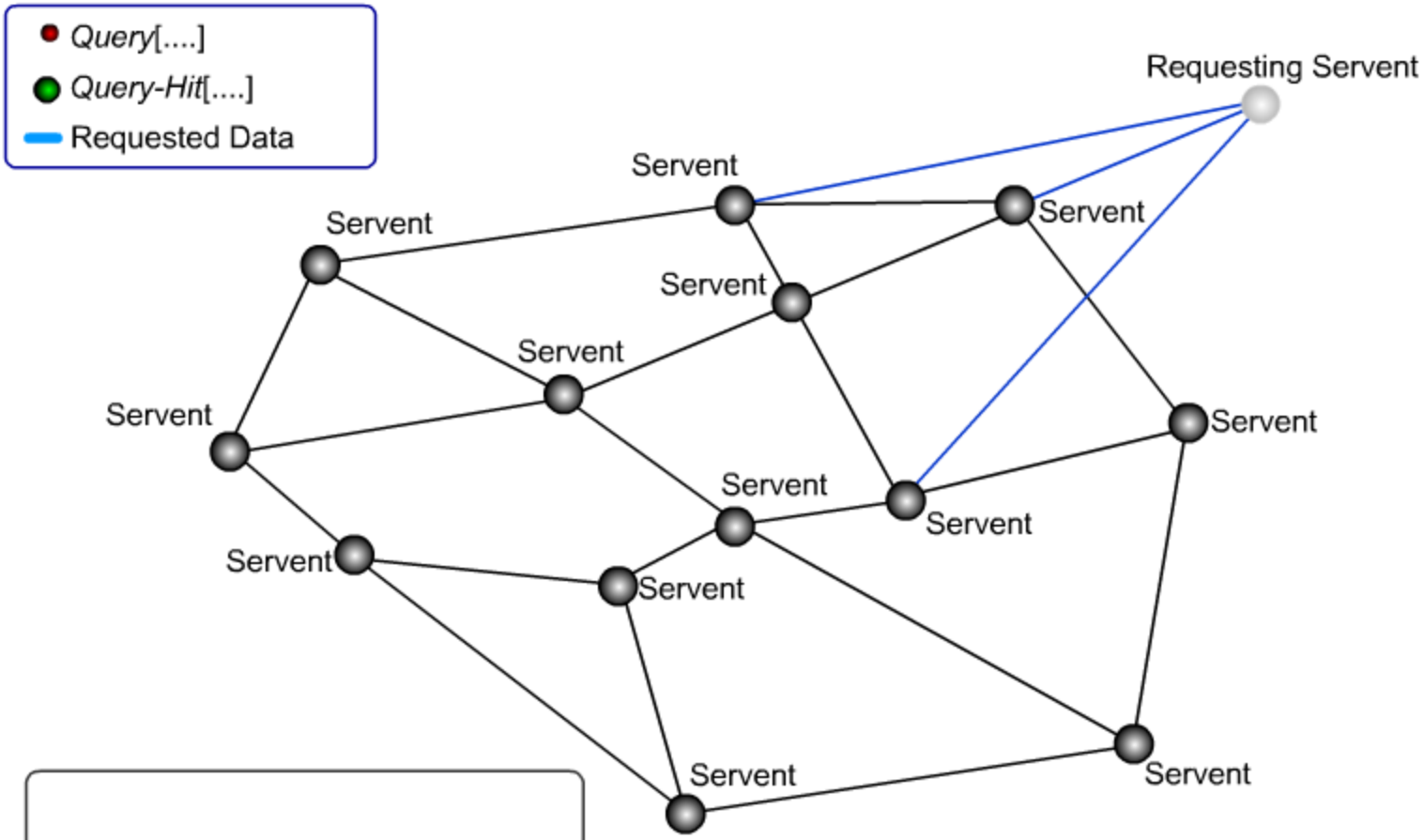
La imposición del TTL impone un horizonte a cada cliente.

P2P Puro – Gnutella: Protocolo

Se basa en cuatro primitivas

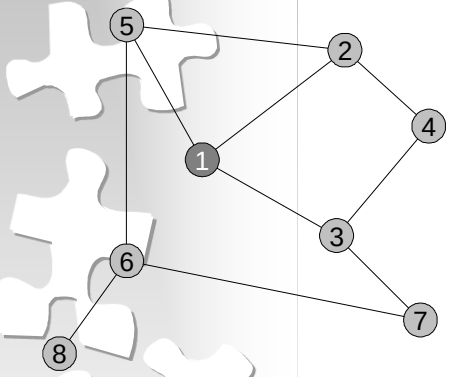
- Ping: Requerimiento de un host para autoanunciarse.
- Pong: Respuesta al Ping. Contiene el IP y el pórtico del host que responde, el número y tamaño de archivos compartidos.
- Query: Un requerimiento de búsqueda. Contiene un string de búsqueda y los mínimos requerimientos de velocidad del host que responde.
- QueryHits: respuesta a Query. Contiene IP, pórtico y velocidad, el número de archivos encontrados y el conjunto de índices resultado.

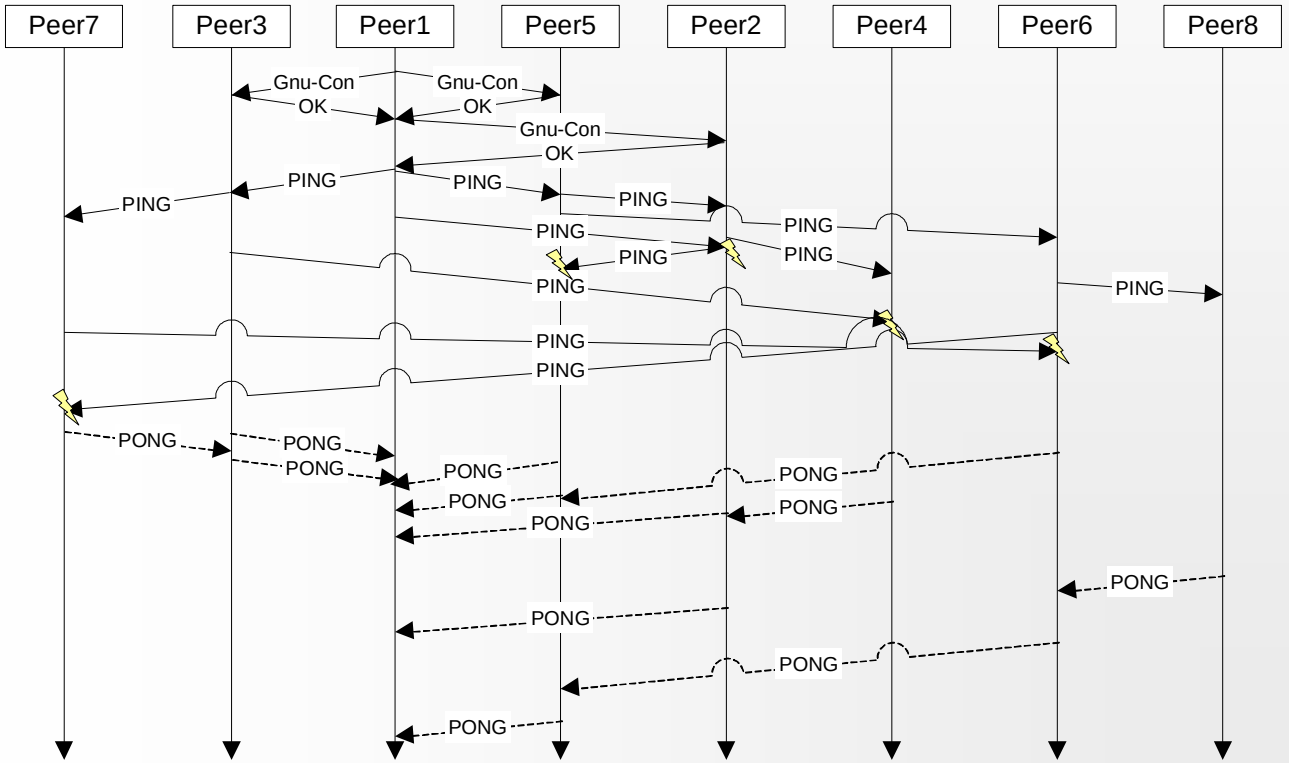
P2P Puro - Gnutella



Sumario de señalamiento en Gnutella 0.4

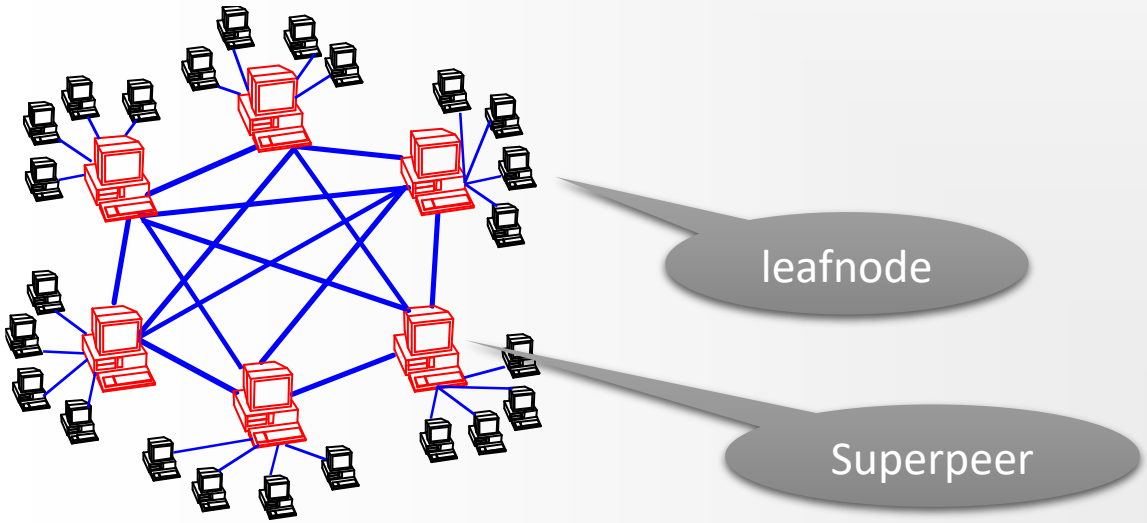
- Ejemplo de protocolo de mensajes de acuerdo con la red del ejemplo



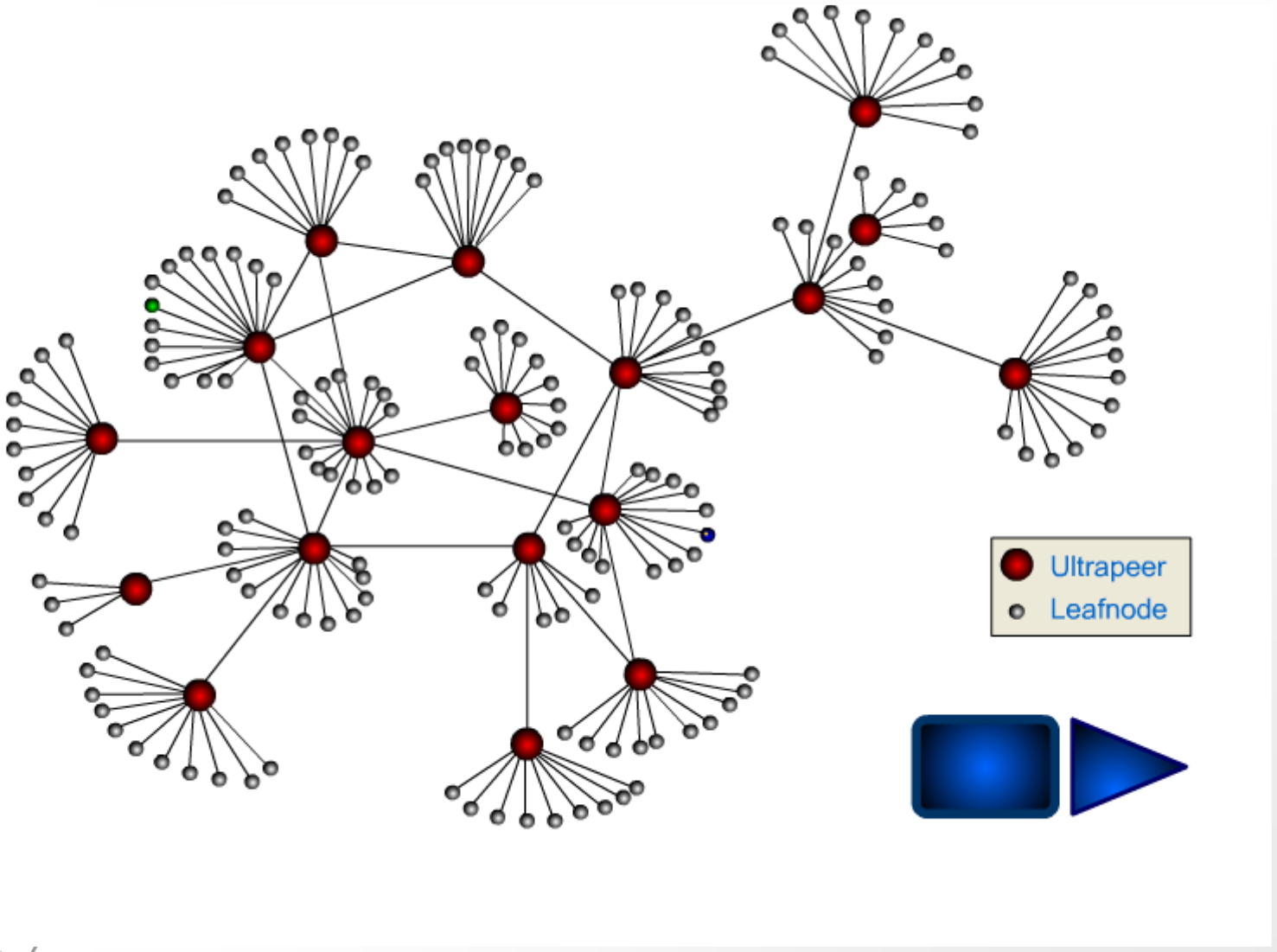


Definición de P2P híbrida

- Principal característica comparada con P2P puro: Introducción de otra capa dinámica jerárquica
- Red basado en hub
- Reduce la carga de señalamiento sin reducir la confiabilidad
- Proceso de elección para seleccionar la asignación de un Superpeer
- Superpeers: alto grado (>>20, dependiendo del tamaño de la red)
- Leafnodes: conectado a uno o más Superpeers (<7)



Gnutella 0.6: Como trabaja



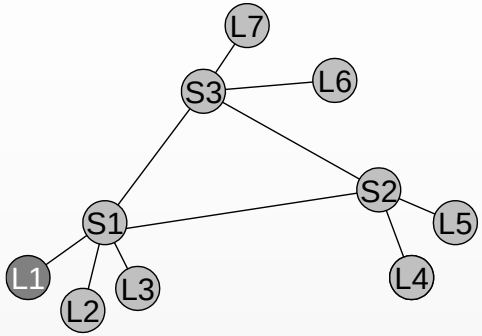
Sistemas P2P basados en P2P híbridos

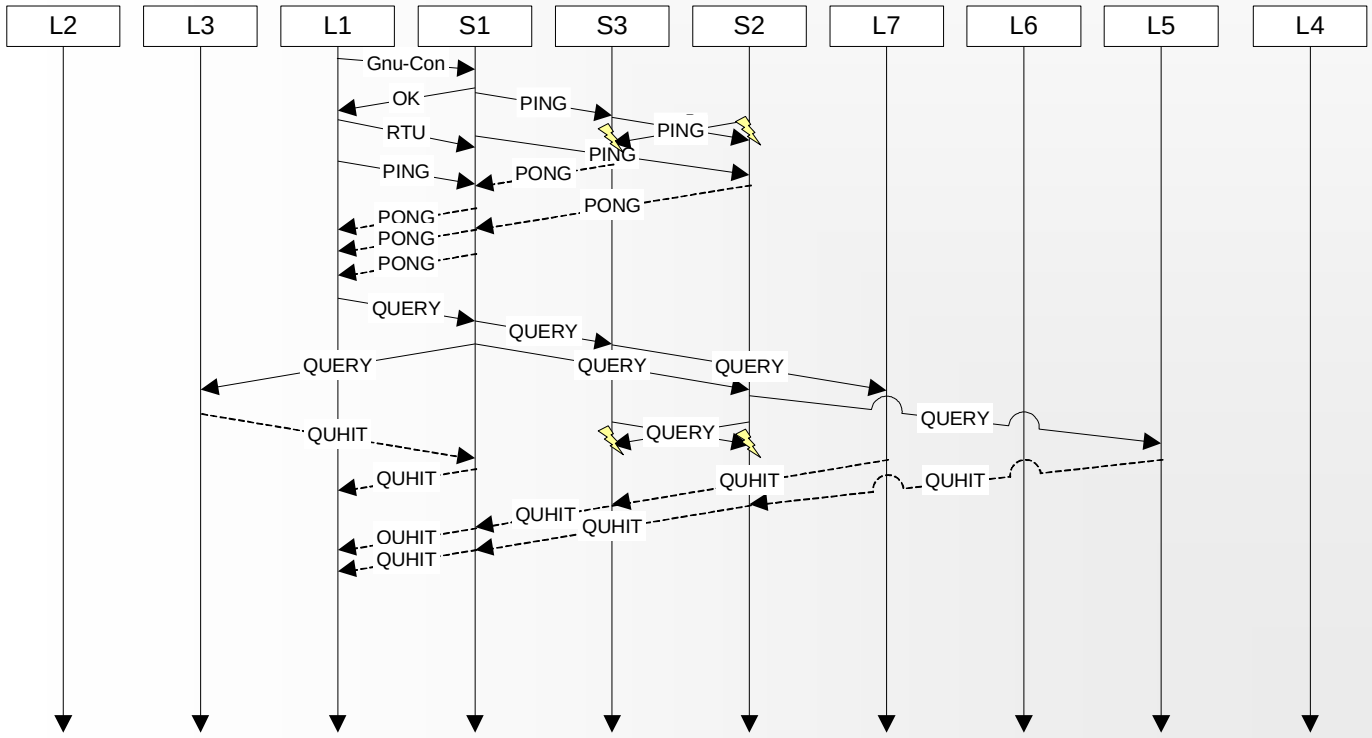
- Edonkey
- Kazaa/FastTrack
- Emule
- OpenNap
- …

Sumario de señalamiento en Gnutella 0.6

- Ejemplo de red Gnutella 0.6

  Secuencia de mensajes de acuerdo al ejemplo de red





Agenda

1. Característica
2. Evolución
3. Aplicaciones
4. Primera, Segunda y Tercera Generación
5. Métodos de Búsqueda. DHT

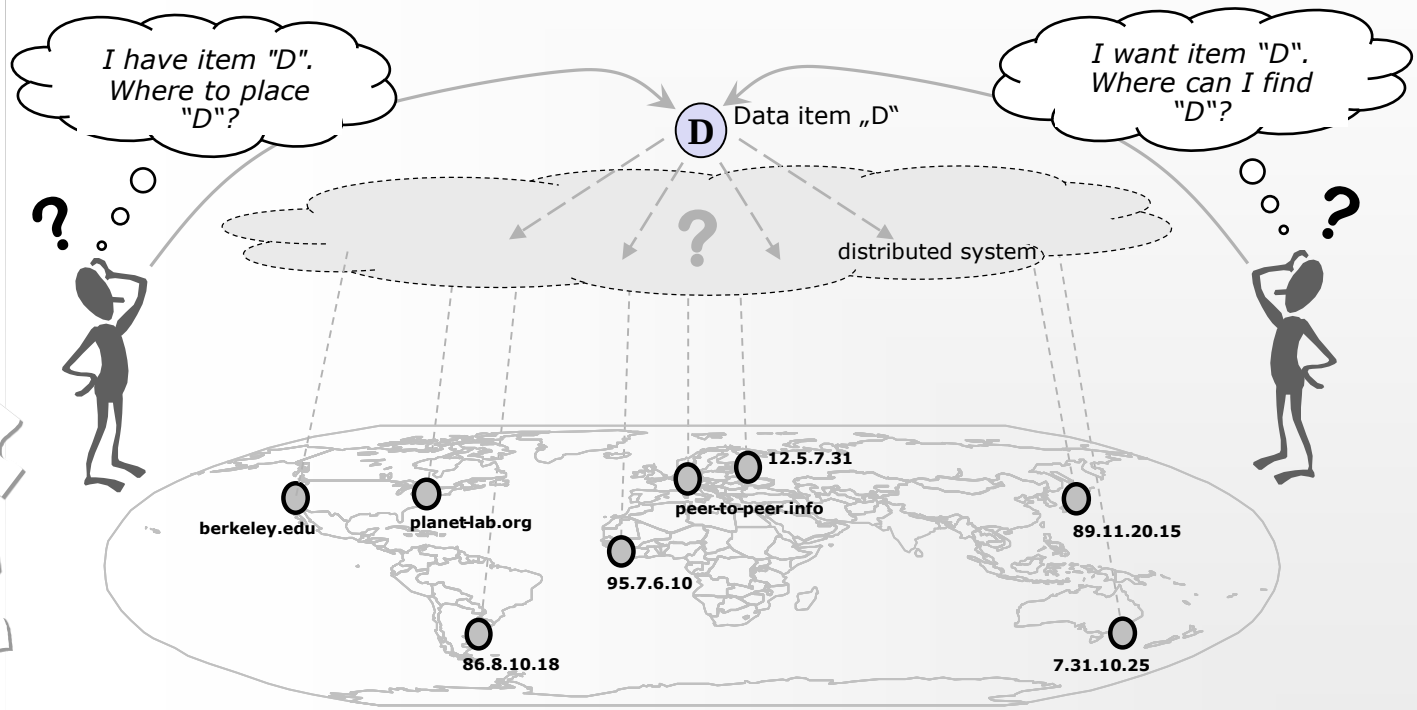
Modelo de ruteo de Documentos

- ***Modelo de ruteo de Documentos:***
  - Sin autoridad central de coordinación (todos los peers son iguales).
  - Los archivos no están almacenados en el HW de los peers que lo proveen.
  - Los archivos está almacenados en otras locaciones de la red P2P.
    - Asigna responsabilidad por un conjunto de archivos a cada peer (de acuerdo a una función definida).
    - Rutea archivos a peers asociados que los almacenan.
    - Se usa una función definida para determinar el peer asociado cuando hay un requerimiento.
    - Se baja el archivo del peer asociado.
  - Usa una función hash: *"Distributed Hash Table (DHT)"*

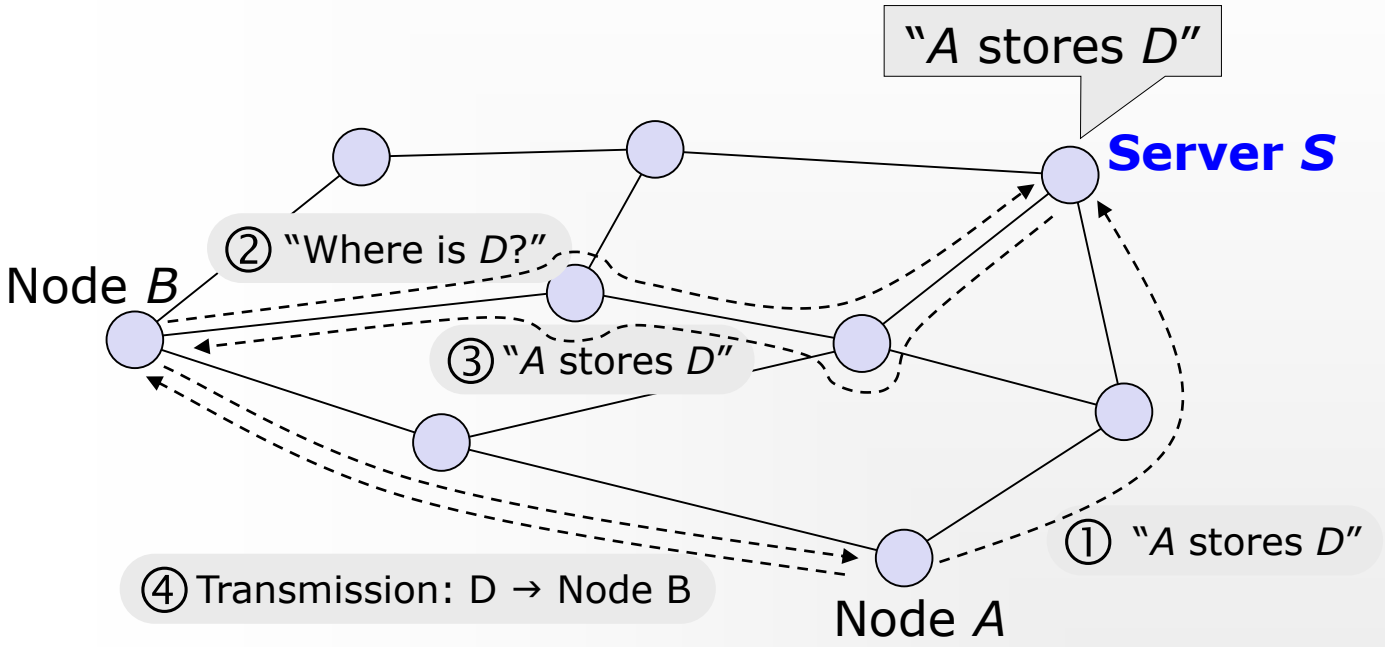
Modelo de ruteo de Documentos

- Características:
  - La autoorganización permite la adaptación en casos de entrada y salida necesaria de peers.
- Ejemplos:
  - *Pastry, Chord, CAN, Tapestry, Freenet.*

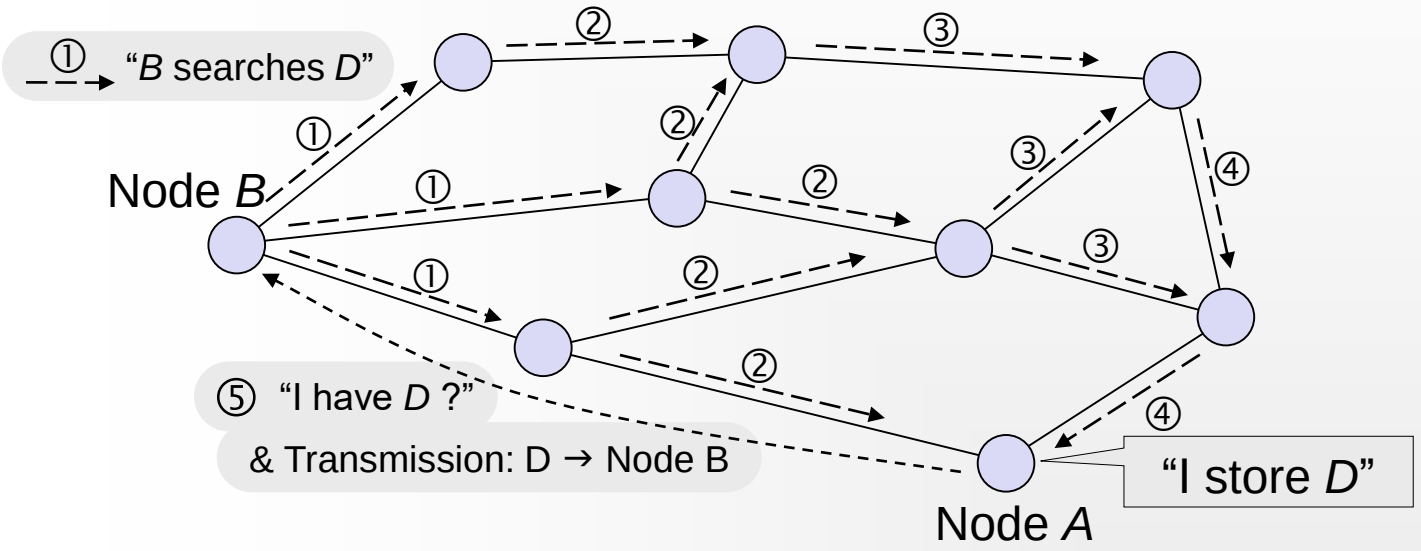
Fundamentos de Búsqueda



Fundamentos de Búsqueda - Centralizada

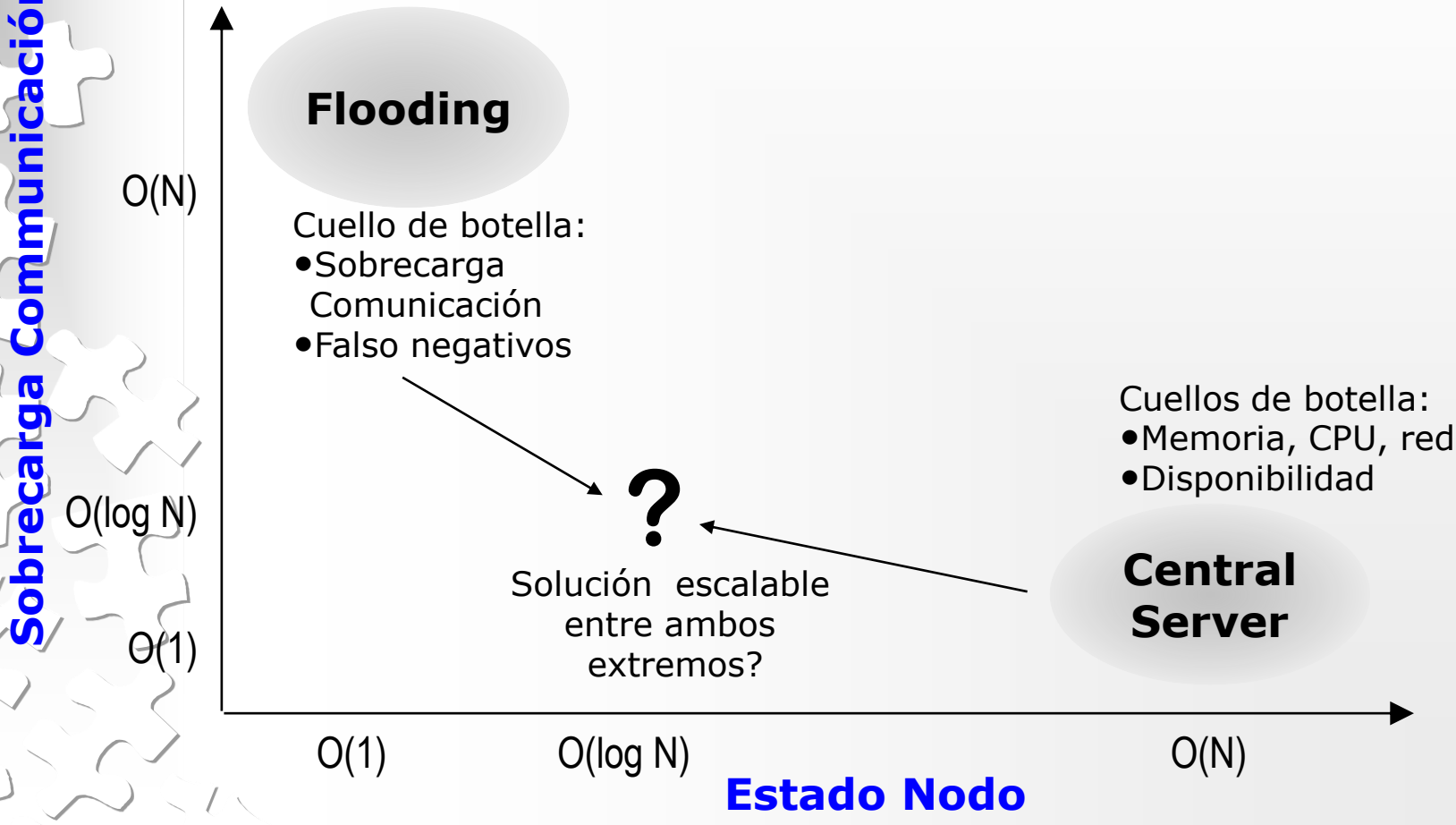


Fundamentos de Búsqueda - Inundación



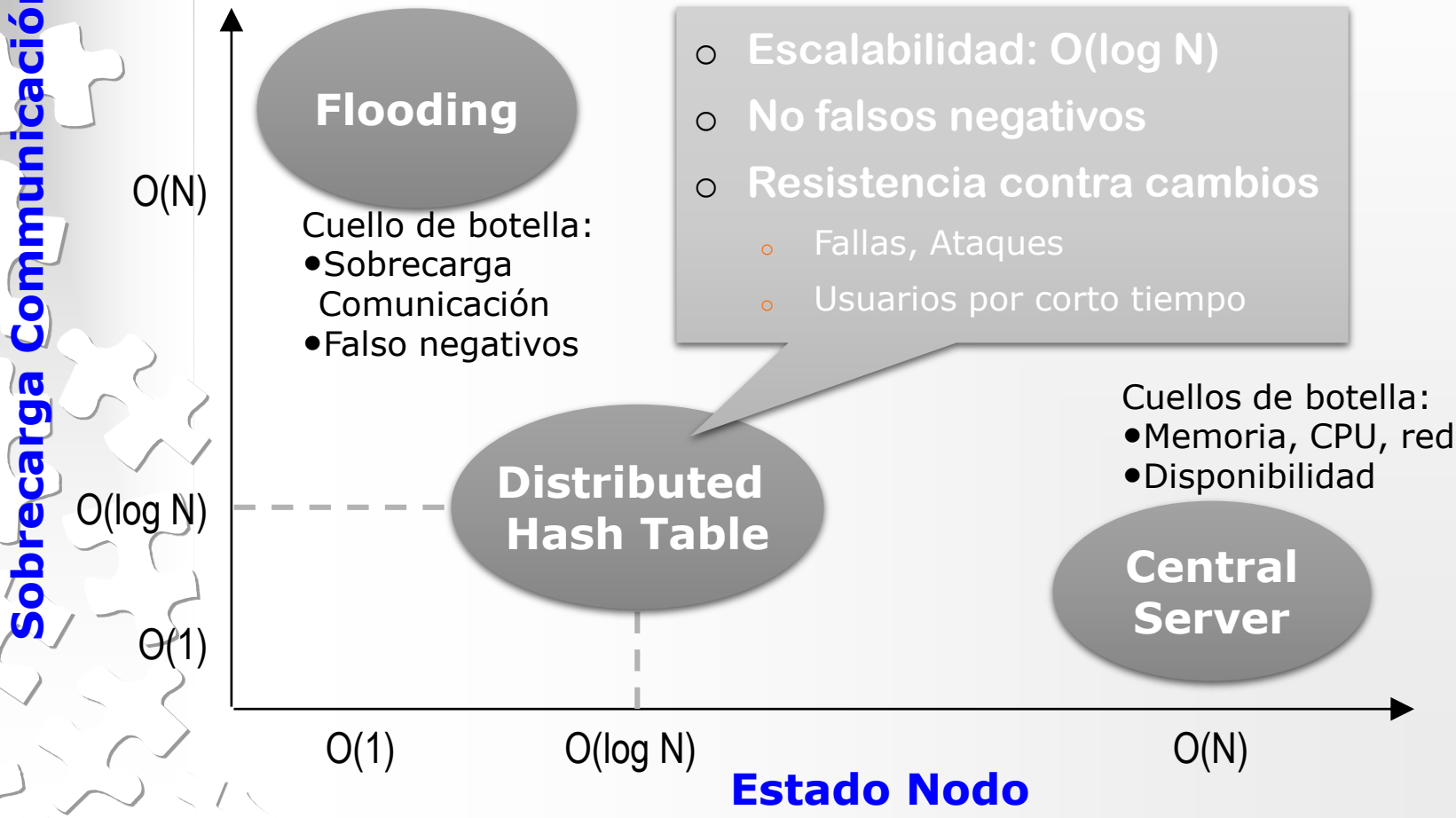
Motivación de Indexado Distribuido

- Sobrecarga de Comunicación vs. estado del nodo

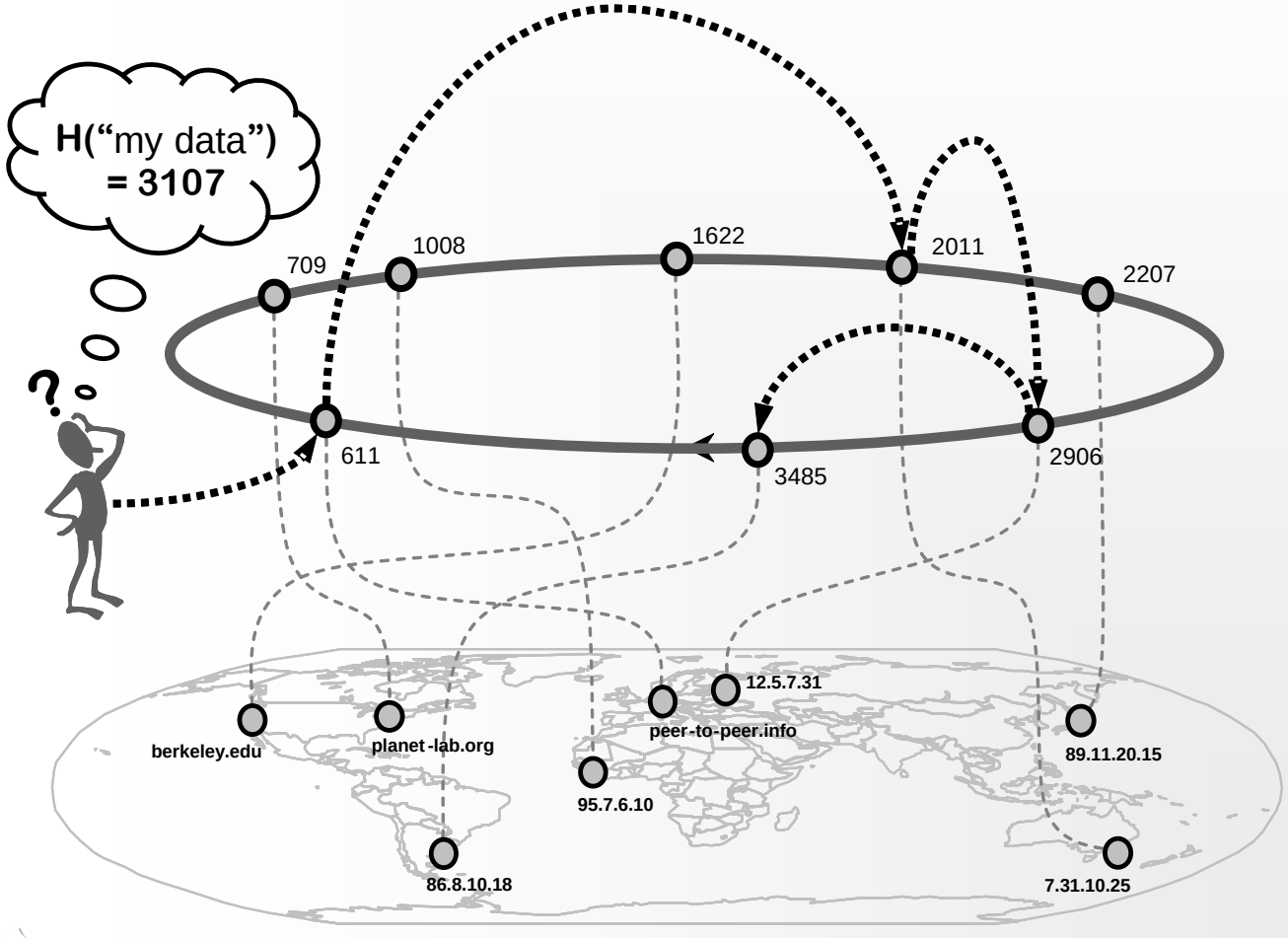


Motivación de Indexado Distribuido

- Sobrecarga de Comunicación vs. estado del nodo



Fundamentos de Búsqueda DHT



Fundamentos de DHT

- Desafíos en el diseño de DHTs
  - Características Deseadas
    - Flexibilidad
    - Confiabilidad
    - Escalabilidad
  - Igual distribución de contenidos entre nodos
    - Crucial para una eficiente búsqueda de contenido
  - Permanente adaptación a fallas, agregados y salidas de nodos
    - Asignación de responsabilidades a nuevos nodos
    - Reasignación y redistribución de responsibilidades en caso de fallas de nodo o partida del mismo

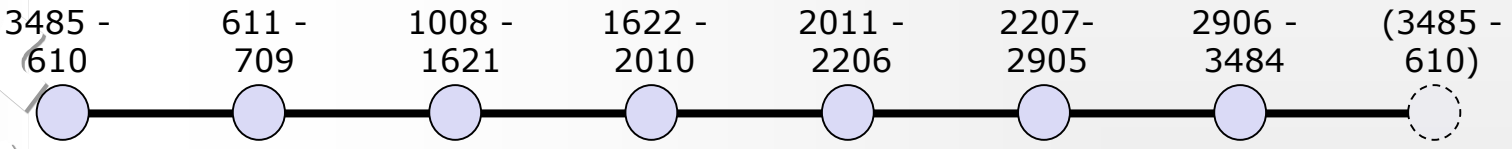
Manejo Distribuido de Datos

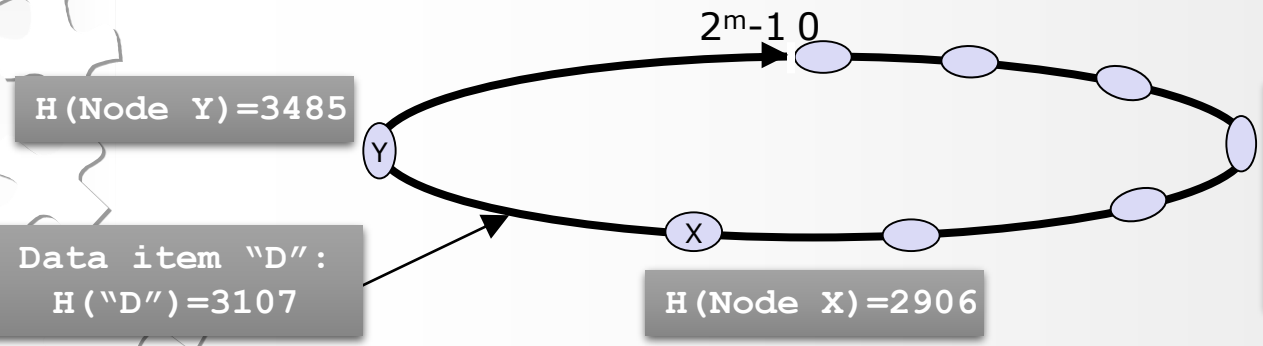
Secuencia de operaciones

1. **Mapeo de nodos y datos en el mismo espacio de direcciones**
   - Peers y contenidos son direccionados usando identificadores flat (IDs)
   - Espacio común de direcciones para datos y nodos
   - Los nodos son responsables por datos en ciertas partes del espacio de direcciones
   - La asociación de datos a nodos puede cambiar dado que los nodos pueden desaparecer
2. **Almacenamiento/Búsqueda de datos en DHT**
   - Búsqueda de datos = ruteo al nodo responsable
     - El nodo responsable no necesariamente es conocido en avance
     - Sentencia determinística acerca de la disponibilidad de datos

Direccionamiento en DHT

- Mapeo de contenido/nodo en espacio lineal
  - Usualmente: 0, …, 2^m-1 >> número de objetos a ser almacenados
  - Mapeo de datos y nodos en el espacio de direcciones (con hash)
    - Ej., Hash(*String*) mod 2^m: H("*my data*") → 2313
  - Asociación de partes de espacio de direcciones a nodos HT





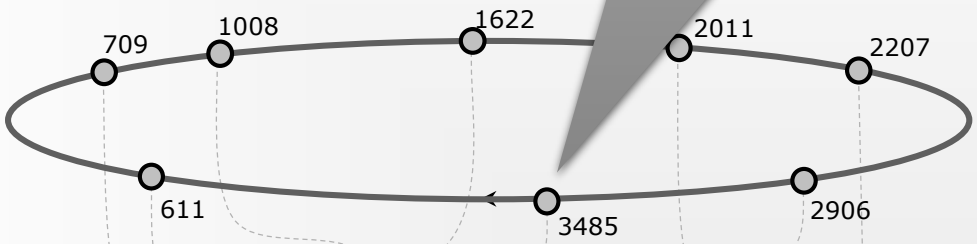
Frecuentemente, el espacio de direcciones es visto como un círculo.

Asociación del espacio de direcciones con nodos

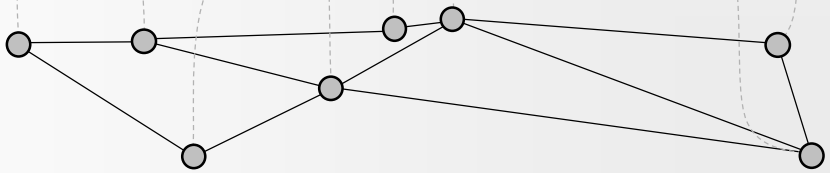
- Cada nodo es responsable por una parte del rango de valores
  - Frecuentemente con redundancia (solapado de partes)
  - Adaptación continua
  - Las topologías real (underlay) y lógica (overlay) no correlacionan

Nodo 3485 es responsable por items de datos en rango 2907 to 3485

Vista lógica de DHT



Mapeo a la topología real

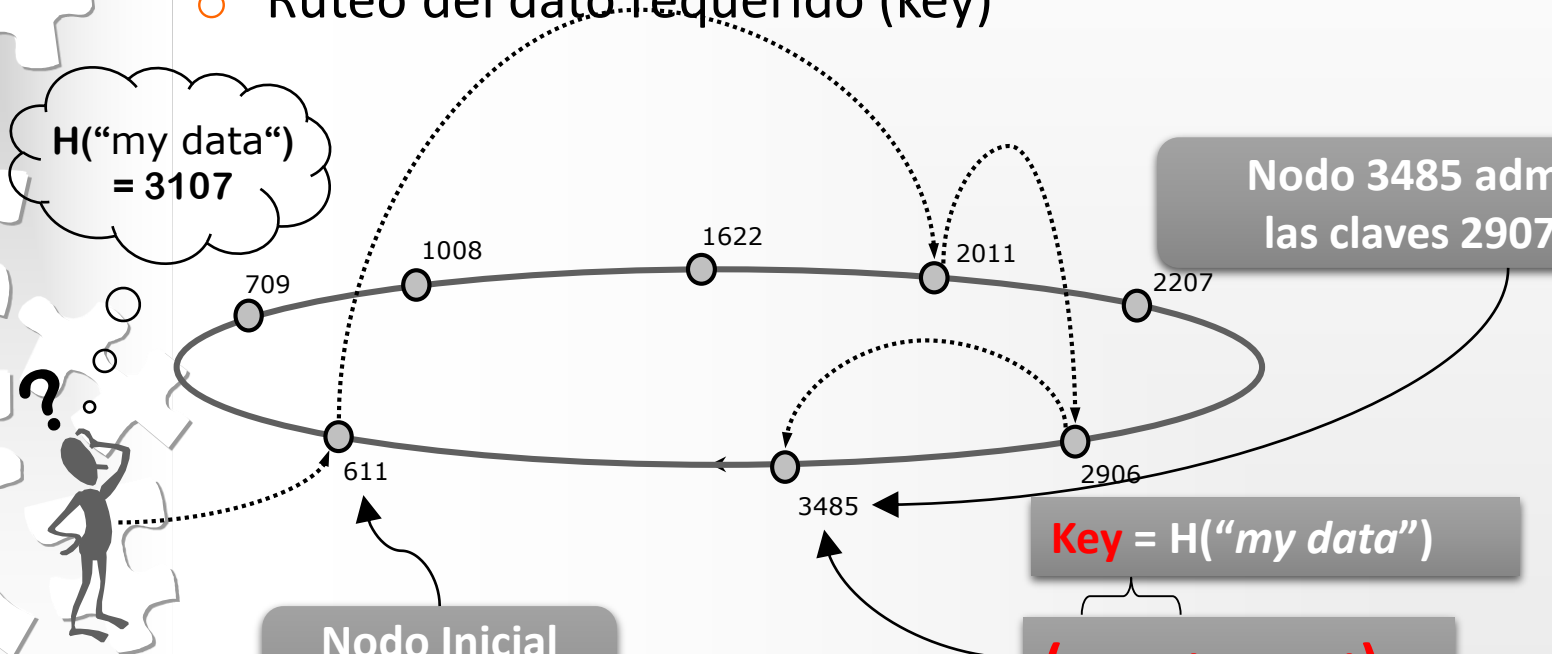


Ruteo de un item de dato

- **Localización del dato** (ruteo basado en contenido)
- Objetivo: Pequeño y esfuerzo escalable
  - O(1) con tabla hash centralizada
    - Pero:
      Su manejo es muy costoso (servidor)
  - Sobrecarga mínima con DHT
    - O(log N): Orden en DHT para localizar un objeto
    - O(log N): número de claves e información de ruteo por nodo (N = # nodos)

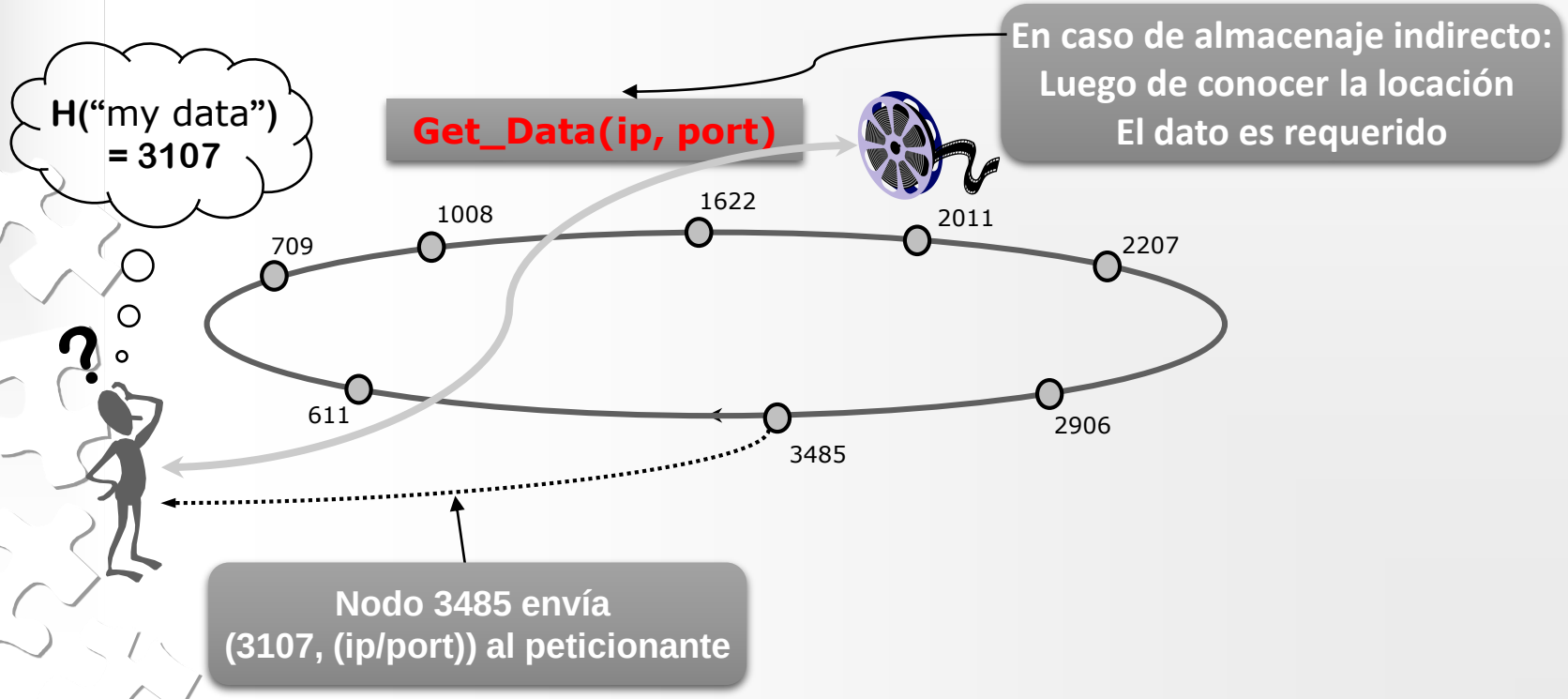
Ruteo de un item de dato

- **Ruteo de un par K/V**
  - Comienza la búsqueda en un nodo arbitrario de DHT
  - Ruteo del dato requerido (key)



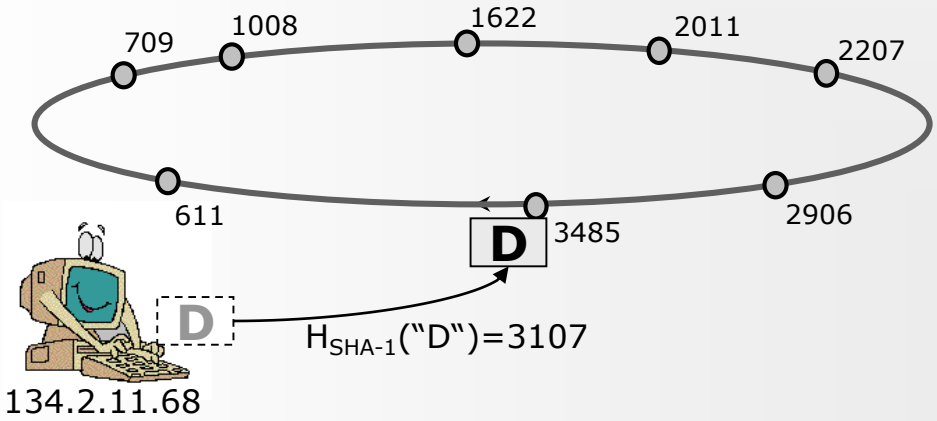
Ruteo de un item de dato

- **Obteniendo el contenido**
  - El par K/V es atendido por el peticionante
  - El peticionante analiza la K/V-tuple (y baja el dato de la actual locación – en caso de almacenaje indirecto)



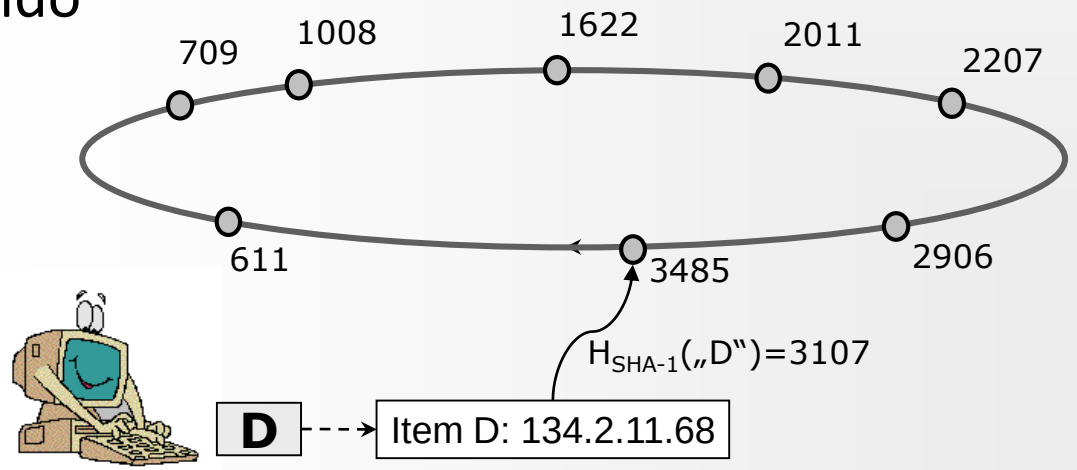
Asociación de datos con IDs – Almacenaje Directo

- ¿Cómo es almacenado el contenido en los nodos?
  - Ejemplo:
    H(“*my data*”) = 3107 es mapeado en el espacio de direcciones DHT
- Almacenaje Directo
  - El contenido es almacenado en el nodo responsable para H(“*my data*”)
  - → **Inflexible** para grandes contenidos – o.k., si los datos son pequeños (<1KB)



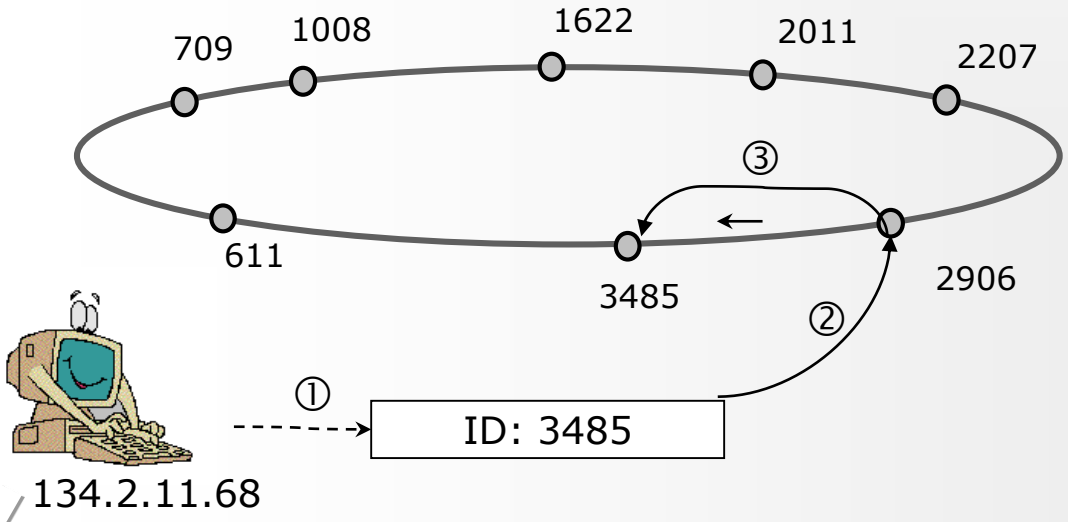
Asociación de datos con IDs – Almacenaje Indirecto

- Almacenaje Indirecto
  - Nodos en una DHT almacenan tuplas como (key,value)
    - Key = Hash(“*my data*”) → 2313
    - Value es una dirección real de almacenaje de contenido: (IP, Port) = (134.2.11.140, 4711)
  - **Más flexible**, pero un paso más para alcanzar el contenido



Llegada de Nodo

- Agregando un nuevo nodo
  1. Cálculo del ID del nodo
  2. El nuevo nodo contacta DHT vía un nodo arbitrario
  3. Le asigna un particular rango hash
  4. Copia de los pares K/V del rango hash (usualmente con redundancia)
  5. Mapeo en el ambiente de ruteo



Falla del Nodo/Partida

- Falla del nodo
  - Uso de pares redundantes K/V (si el nodo falla)
  - Uso caminos de ruteo redundantes/ alternativos
  - El par Key-value usualmente continua recuperable en al menos una copia

- Partida de un nodo
  - Particionado del rango hash entre los nodos vecinos
  - Copia de los pares K/V en los nodos correspondientes
  - Se quita del ambiente de ruteo

DHT Interfaces

DHT Ejemplo - Pastry

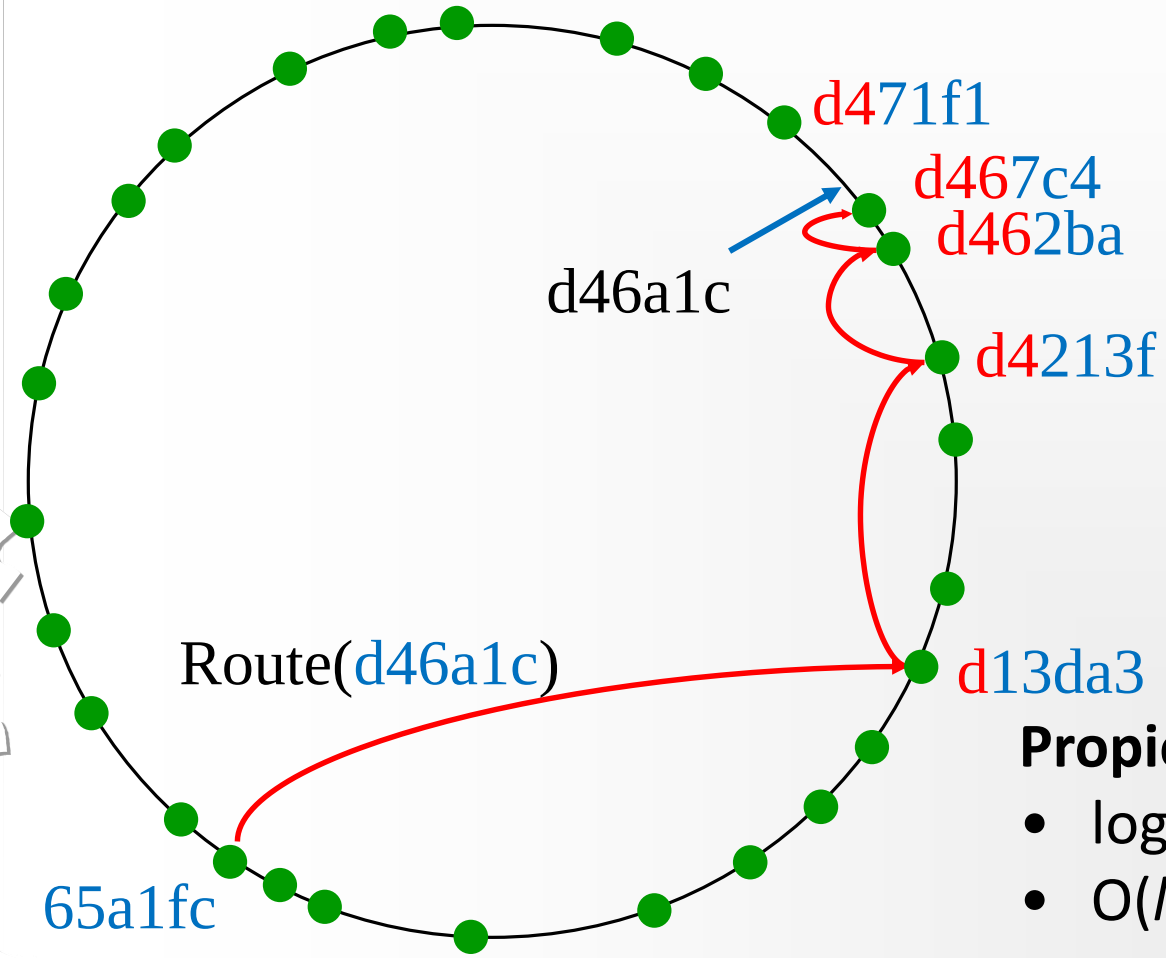
- Asigna los nodos y objetos a números b-arios de m dígitos.
- Asigna cada objeto al nodo con el que comparte el prefijo más grande.
- Por ejemplo:
  - b = 4 and m = 6

DHT Ejemplo - Pastry Tabla Ruteo (# 65a1fcx)

Fila 0	**0x**	**1x**	**2x**	**3x**	**4x**	**5x**		**7x**	**8x**	**9x**	**ax**	**bx**	**cx**	**dx**	**ex**	**fx**
Fila 1	60x	61x	62x	63x	64x		66x	67x	68x	69x	6ax	6bx	6cx	6dx	6ex	6fx
Fila 2	650x	651x	652x	653x	654x	655x	656x	657x	658x	659x		65bx	65cx	65dx	65ex	65fx
Fila 3	65a0x		65a2x	65a3x	65a4x	65a5x	65a6x	65a7x	65a8x	65a9x	65aax	65abx	65acx	65adx	65aex	65afx

$\log_{16} N$ filas

DHT Ejemplo - Pastry Ruteo



Propiedades

- $\log_{16}$ N pasos
- O(*log N*) estado

DHT Ejemplo - Chord

Principales características

- Enrutamiento (Routing)
  - Espacio de direcciones lógicas planas: identificadores de l bits en lugar de direcciones IP
  - Enrutamiento eficiente en sistemas grandes: log(N) saltos con N nodos totales
- Autoorganización

Manejar la llegada, salida y falla del nodo

DHT Ejemplo - Chord

- Enrutamiento primitivo:
  Consulta hacia adelante por la clave x hasta que se encuentre el sucesor (x)
  Devuelve el resultado a la fuente de consulta
- Pros:
  Simple
  Pequeño estado del nodo
- Contras:
  Mala eficiencia de búsqueda: O (1/2 * N) saltos en promedio (con N nodos)
  La falla del nodo rompe el círculo

DHT Ejemplo - Chord

Tabla de ruteo de Chord – Tabla Finger

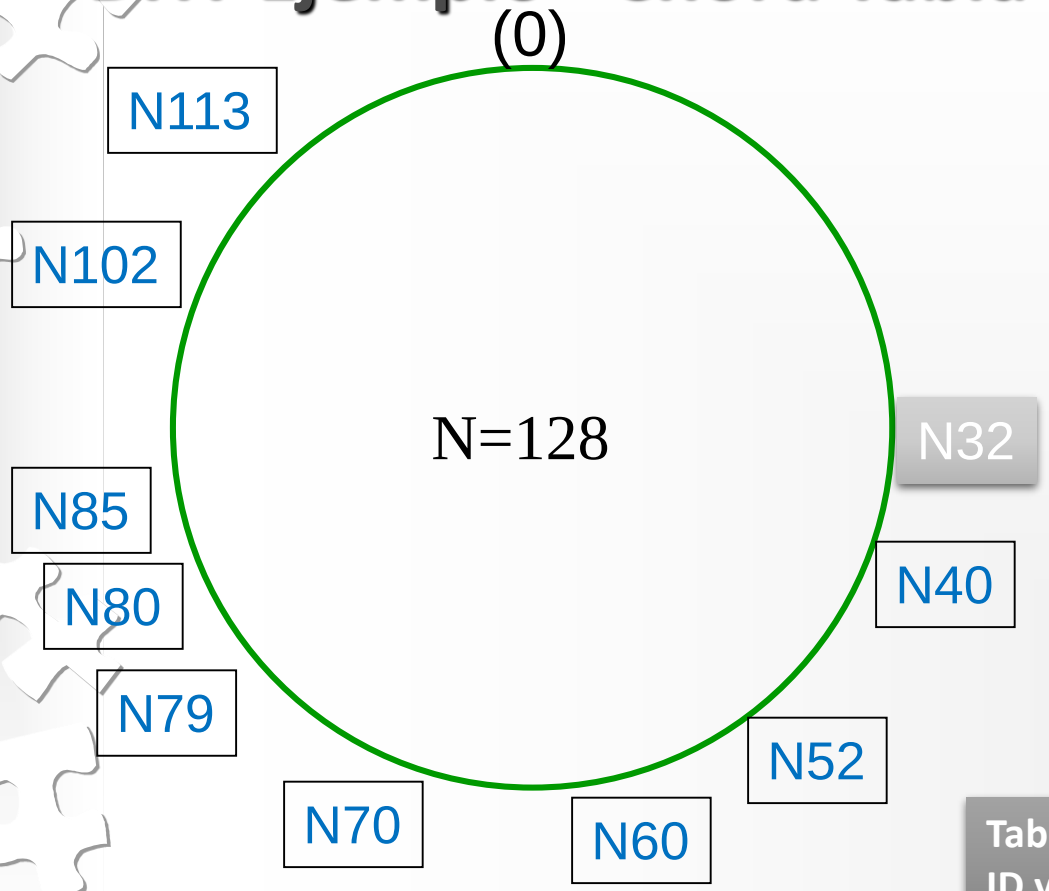
- Almacena log(N) enlaces por nodo
- Cubre distancias que aumentan exponencialmente: Nodo n: la entrada i apunta al sucesor (n + 2 ^ i) (i-ésimo elemento de la tabla Finger)

DHT Ejemplo - Chord

Algoritmo de ruteo Chord

- Cada nodo n consulta hacia adelante por la clave k en sentido horario
  - Al nodo más lejano que precede a k
  - Hasta que n = predecesor (k) y sucesor (n) = sucesor (k)
  - Devuelve el sucesor (n) a la fuente de consulta

DHT Ejemplo - Chord Tabla Finger



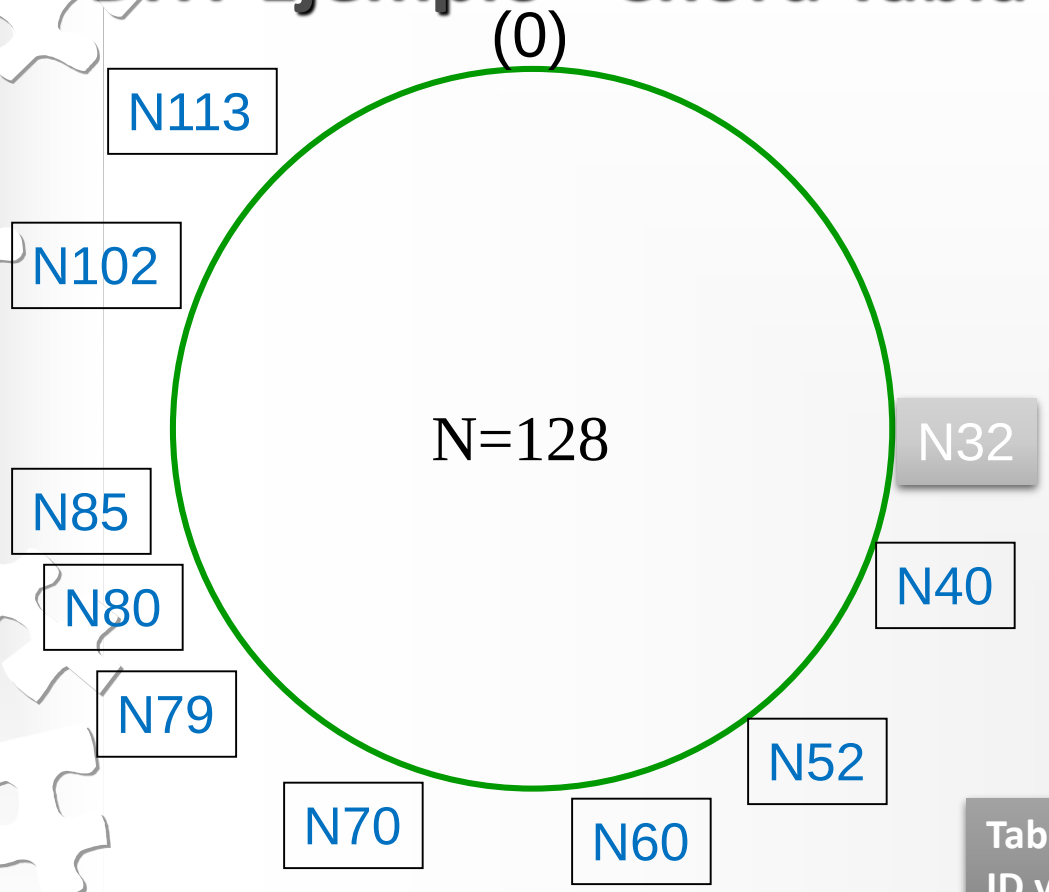
N32's Tabla Finger

Id	Target ID	Sucessor
0	N32 + 1	N40
1	N32 + 2	N40
2	N32 + 4	N40
3	N32 + 8	N40
4	N32 + 16	N52
5	N32 + 32	N70
6	N32 + 64	N102

Tabla Finger contiene actualmente ID y dirección IP

Nodo n's i-ésima entrada: primer nodo $\geq n + 2^{i-1}$

DHT Ejemplo - Chord Tabla Finger



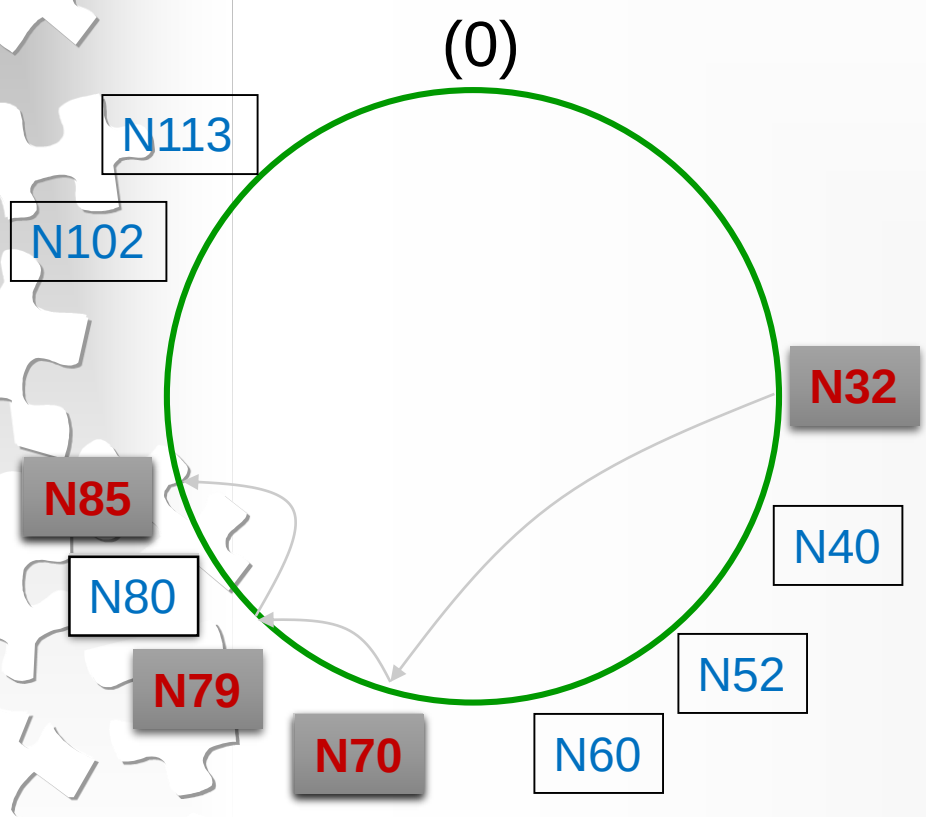
N32's Tabla Finger

33..33	N40
34..35	N40
36..39	N40
40..47	N40
48..63	N52
64..95	N70
96..31	N102

Tabla Finger contiene actualmente ID y dirección IP

Nodo n's i-ésima entrada: primer nodo $\geq n + 2^{i-1}$

DHT Ejemplo - Chord Búsqueda



N32's Tabla Finger

33..33	N40
34..35	N40
36..39	N40
40..47	N40
48..63	N52
64..95	**N70**
96..31	N102

N70's Tabla Finger

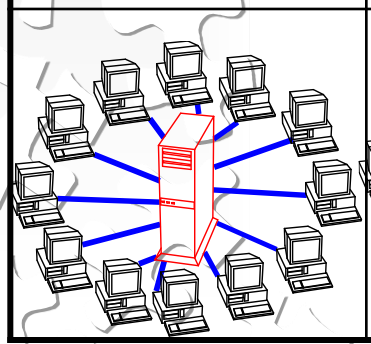
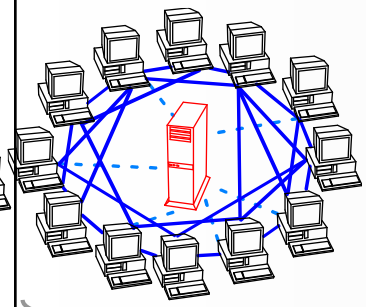
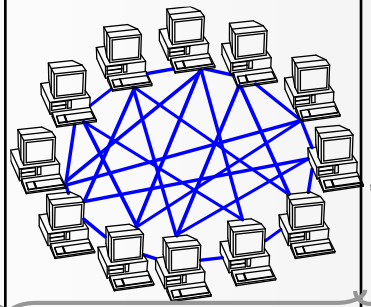
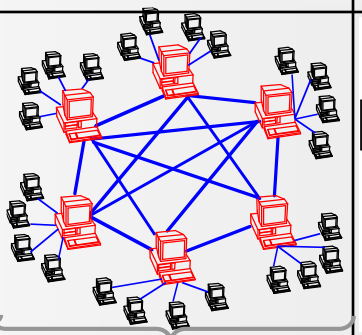
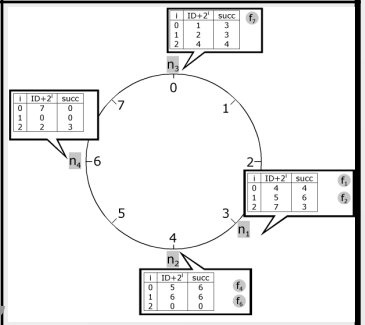
71..71	N79
72..73	N79
74..77	N79
78..85	**N79**
86..101	N102
102..5	N102
6..69	N32

N79's Tabla Finger

80..80	N80
81..82	**N85**
83..86	N85
87..94	N102
95..110	N102
111..14	N113
15..78	N32

Node 32, búsqueda(82): 32 → 70 → 79 → 85.

Resumen de Arquitecturas de la Primera y Segunda Generación de P2P

Cliente-Servidor	***Peer-to-Peer***			
1. El Servidor es la entidad central y el único proveedor del servicio y contenido. → La red es manejada por el servidor 2. El Servidor es el sistema de más alto rendimieno. 3. El Cliente es el sistema d más bajo rendimiento Ejemplo: WWW	1. Los recursos son compartidos entre peers 2. Los recursos pueden ser accedidos directamente desde otros peers 3. Concepto de Servent			
	P2P sin Estructura			***P2P Estructurado***
	P2P Centralizado	***P2P Puro***	***P2P Híbrido***	***Basado en DHT***
	1. Todas las características de P2P incluidas 2. Una entidad central es necesaria para proveer el servicio 3. La entidad central es una especie de Db índice/grupo Ejemplo: Napster	1. Todas las características de P2P incluidas 2. Alguna entidad terminal puede ser removida sin pérdida de funcionalidad 3. → Entidades no central es Ejemplos: Gnutella 0.4, Freenet	1. Todas las características de P2P incluidas 2. Alguna entidad terminal puede ser removida sin pérdida de funcionalidad 3. → Enidades centrales dinámicas Ejemplo: Gnutella 0.6, JXTA	1. Todas las características de P2P incluidas 2. Alguna entidad terminal puede ser removida sin pérdida de funcionalidad 3. → Entidades no central es 4. Las conexiones en overlay son “fijas” Ejemplos: Chord, CAN
				

1era Gen. **2da Gen.**

Blockchain

- Artículo para leer: An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. https://www.researchgate.net/publication/318131748

Bibliografía:

- Steinmetz, R y Wehrle, K.; “Peer-to-Peer Systems and Applications”. Springer, 2005.
- Coulouris,G.F.; Dollimore, J. y T. Kindberg; “Distributed Systems: Concepts and Design”. 5th Edition Addison Wesley, 2011.
- S. Androutsellis-Theotokis, D. Spinellis; A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys, #4, vol 36, dec 2004.