

P2P Networks

Introduction

Pedro García López

Universitat Rovira I Virgili

Pedro.garcia@urv.net

Departament d'Enginyeria



Informàtica i
Matemàtiques



UNIVERSITAT
ROVIRA I VIRGILI

Index

- Goals
- Professor Information
- Course Outline
- Evaluation
- Introduction to P2P Networks

Goals

- It is an optional course focused on recent and ongoing research work in P2P networks.
- It is a seminar course and active class-room and laboratory participation is expected. Each student is required to read and present research papers.
- The course is designed to encourage everyone to actively learn advanced concepts, to independently think over research and development issues, to pro-actively relate what we learn to the real problems in practice, to stimulate and brain-storm new ideas, to intelligently solve pressing problems in various phases of P2P technologies.
- A key goal is to understand key P2P algorithms and infrastructures, and to test them in simulation (PlanetSim Overlay simulator) and real network settings (PlanetLab).

Professor Information

- Pedro García Lopez, PhD
 - Office 238, ETSE, DEIM
 - Mail: pedro.garcia@urv.net
 - Web: <http://www.etse.urv.es/~pgarcia/>
 - MSN: tutorias_etse@hotmail.com
- Marc Sanchez Artigas, PhD Student
 - Lab 144
 - Mail: marc.sanchez@urv.net
 - Web: <http://ants.etse.urv.es/marc/>

Professor timetable

- Office 238
 - Monday: 16:00 -19:00
 - Tuesday: 12:00 -13:00 & 16:00 – 19:00
- MSN
 - tutorias_etse@hotmail.com

Evaluation

No final exam !!

- Task 1: Simulator problem assignment
- Task 2: 20 minutes presentation and study of a selected research paper
- Task 3: Final assignment to select between
 - Development of a P2P application (proposed by professor or students)
 - Research assignment: algorithm study in the simulator
 - P2P technology evaluation in the PlanetLab network (Emule, Gnutella, Bittorrent)
- Misc: Two or three Moodle tests

Note: Many lab hours will be available for developing the assignments.

Note: Attending to classroom is not mandatory. Everything will be in MOODLE (distance learning).

Course Outline

1. Introduction to P2P Networks
2. Unstructured and Hybrid Networks
3. Structured Networks (1)
4. Structured Networks (2)
5. Hierarchical Networks and Topology aware overlays
6. P2P content Search and retrieval
7. Security in P2P networks
8. Applications and P2P Middleware platforms

Course Planning

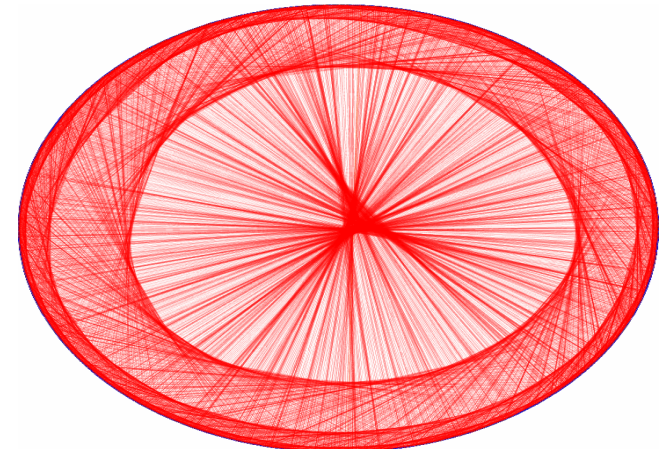
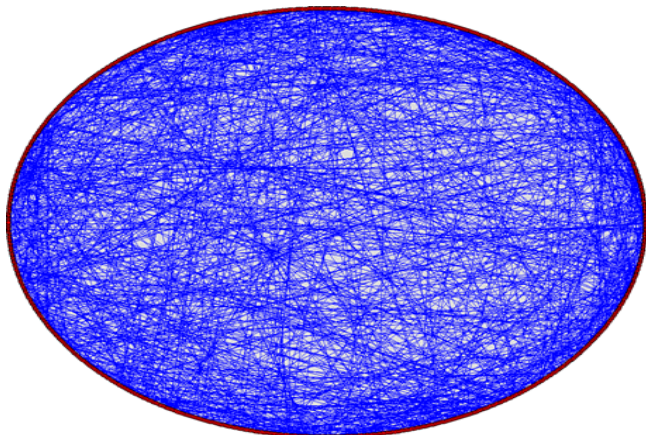
1. Introduction to P2P Networks
2. Unstructured and Hybrid Networks
3. Structured Networks (1)
4. Structured Networks (2) (Task 1)
5. Hierarchical Networks and Topology aware overlays
6. P2P content Search and retrieval
7. Security in P2P networks
8. Applications and P2P Middleware platforms
9. -> 15 Student presentations (Task2)

Task 3: End of the course

Tools: PlanetSim



- Overlay network simulator
- Implemented in Java
- Documentation and tutorials in Labs.
- Layered architecture (decouple p2p protocols from applications)
- Advanced visualization output (Pajek, GML)



PlanetLab optional task 3

- PlanetLab Worldwide Network
- 513 nodes over 240 sites
- Linux machines



Task 3: P2P Applications

- Possible Platforms:
 - Java, FreePastry Implementation
 - Java JXTA
 - Java Gnutella
 - Java Bamboo
 - C++ Symphony Implementation
 - C# Chord Implementation
 - Python Chord Implementation
 - [Your preferred language] [Your protocol]

Task 3 Research Task

- Supervised research tasks (simulation):
 - Ant algorithms
 - Unstructured networks
 - Less structured networks
 - CAST protocols
 - Range queries
 - Protocol comparisons
 - Protocol Implementation
 - Simulator Improvements
 - Proximity network overlays
 - Hierarchical networks

Bibliography

- Peer-to-peer-Systems and Applications. Ralf Steinmetz. Springer LNCS 3485.
- Peer-to-peer, Harnessing the Benefits of disruptive technology. Andy Oram. O'Reilly.
- Sistemas Distribuidos. Conceptos y Diseño. George Colouris. Addison Wesley

Interesting links

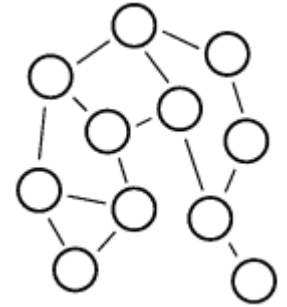
- [P2P Readings](#)
- [DHT links](#)
- [IPTPS](#) Conferences
- [IEEE P2P](#) Conferences
- [IEEE ICDCS](#) Conferences
- [ACM SIGCOMM](#) Conferences
- Other conferences: SIGCOM, INFOCOM, NSDI, ...

Introduction to P2P

- Student A: OOH, but I thought Xarxes P2P Course was about downloading files from Emule !!
- Sorry, this course is about distributed systems ;-)

What is Peer 2 Peer ?

- A distributed system architecture
 - Decentralized control
 - Typically many nodes, but unreliable and heterogeneous
 - Nodes are symmetric in function
 - Take advantage of distributed, shared resources (bandwidth, CPU, storage) on peer-nodes
 - Fault-tolerant, self-organizing
 - Operate in dynamic environment, frequent join and leave is the norm

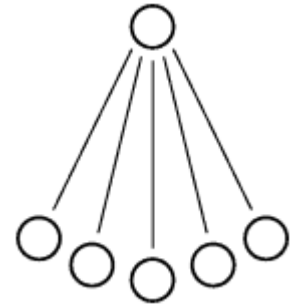


P2P Characteristics

- Decentralization
 - Scalability -> avoid hotspots and bottlenecks!
 - Ad-hoc connectivity
 - Self-organization
 - Local information
 - Fault-resilience
-
- And also: anonymity, security, trust, replication, caching, coordination, ...

What is NOT P2P ?

- A centralized system
 - Client/Server Architecture
 - Master/Slave models
 - Dumb terminals / Active broadcaster
- Problems:
 - Scalability
 - Bottleneck (hotspot)
 - Cost



Relevance of P2P

- According to several internet service providers, more than 50% of Internet traffic is due to P2P applications, sometimes even as much as 75%.
- It is a paradigm shift from coordination to **cooperation**, from centralization to **decentralization**, and from control to **incentives**

Why P2P is important (for engineers)?

- Workstations have considerable computing power (Moore's Law)
- Workstations provide huge data storage capacity
- The available bandwidth is increasing steadily
- Edge computing: benefit from the power at the edges of Internet
- Decentralized applications can scale better (millions of users)
- **Nirvana**: Low cost multiuser applications with endless resources

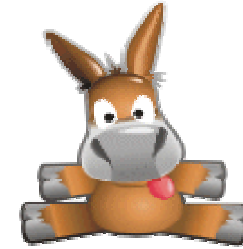
Why P2P is important (for the society)?

- Organizational shift from hierarchical / centralized models to decentralized / network models
- “The Rise of the Network Society”, Manuel Castells
- P2P can enable the construction of worldwide communities
- Example 1. Teaching: classical lecture/exam model Vs collaborative learning
- Example 2. Mass media: big and few broadcasters (TV, radio) Vs public user publishing (Web blogs, net radio, net TV)

P2P Samples

- P2P killer applications:

- USENET (NNTP)
- Instant Messaging (?)
- Napster, Gnutella, eMule, Filetopia, Blubster (MP2P)
- [SETI@HOME](#), [LHC@home](#) (BOINC)
- Groove (Collaboration), Ray Ozzie



- Buzzword: Grid Computing

- OGSA(Open Grid Services Architecture)
- Like P2P, it aims to leverage peer computing resour
- Widespread acceptance in many companies



P2P samples

- *Problems !*

- *[Slashdot 19/01/05]* A Californian bill introduced last week would, if passed, expose file-swapping software developers to fines of up to \$2,500 per charge, or a year in jail, if they don't take 'reasonable care' to prevent their software from being used to commit crime
- *P2P Worms and viruses*

- *Good samples*

- *Skype !*
- *BitTorrent*
- *Peercast (P2P Radio)*
- *Share It! by Bringing P2P into the TV-Domain*



Ideas !!

- P2P technologies will enable a next generation of innovative killer applications.
- P2P and collaboration technologies will promote the creation of worldwide user communities (social networks)
- P2P will permit low cost sophisticated content publishing and distribution tools (net radio, net tv)

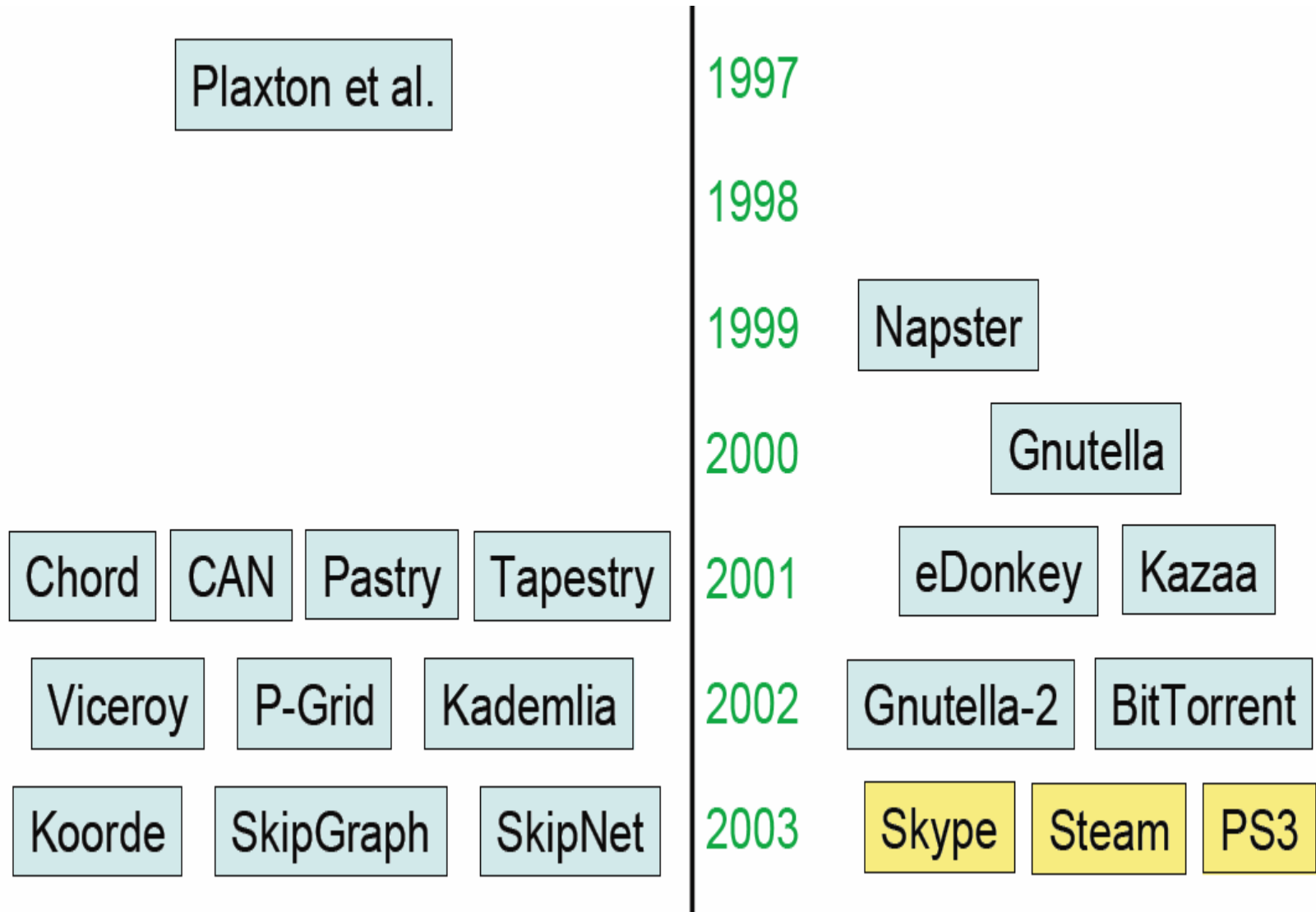
History of P2P: prehistory

- 1969- ARPANET
 - First killer applications like FTP and Telnet were Client/server
- 1979- USENET
 - UUCP, NNTP
 - Decentralized control, avoid network flood, path header, ...
- DNS – 1983
 - Scalability !!!
 - Hierarchical Design
 - Distributed load
 - Caching
 - Query delegation

Internet Explosion (1995-1999)

- The switch to client/server
 - HTTP, Chat, Mail, IM
- Breakdown of cooperation
 - Spam, TCP rate equation, congestion
- Firewalls, dynamic IP, NATs
- Asymmetric Bandwidth

History of P2P

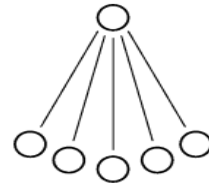


P2P Infrastructures

- Overlay Network:
 - A network on top of a network (IP)
 - Application level routers working in top of an IP network infrastructure
 - Redundancy ?
- Why Overlay Networks ?
 - Focused on application-level services that underlying layers do not offer
 - One-to-Many & Many-to-Many apps, IP Multicast is not supported by most Internet routers
 - IP substrate is difficult to change (IPv6 transition)

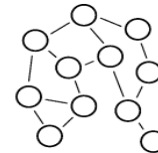
P2P Architectural models

- Centralized (Napster)

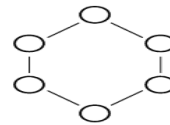


- Decentralized

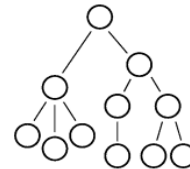
- Unstructured (Gnutella)



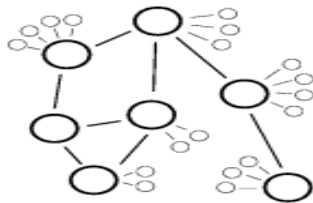
- Structured (Chord)



- Hierarchical (MBone)

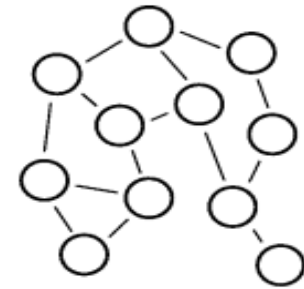


- Hybrid (EDonkey)



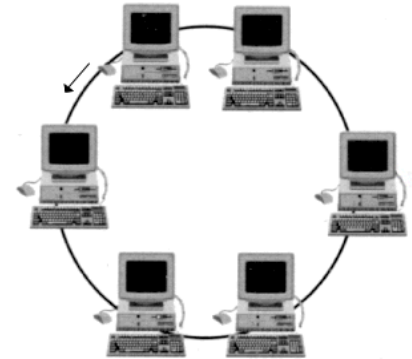
Unstructured P2P Networks

- Gnutella, JxTA, FreeNet
- Techniques:
 - Flooding
 - Replication & Caching
 - Time To Live (TTL)
 - Epidemics & Gossiping protocols
 - Super-Peers
 - Random Walkers & Probabilistic algorithms



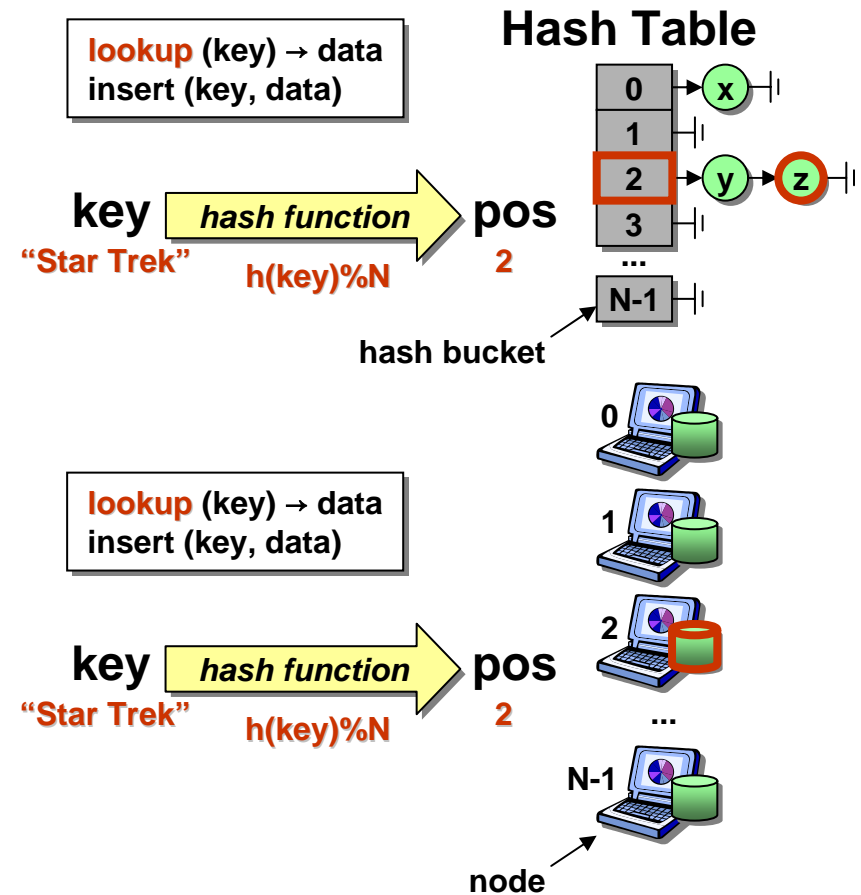
Structured P2P Networks

- Nodes form a regular topology:
 - Chord (Ring)
 - Kademlia (Tree)
 - CAN (Hypercube)
- Assure routing performance to locate contents **$O(\log n)$**
- Every node has a unique id
 - Distance metric over Ids
- Every node has a routing table with links to other nodes
- They are also called Distributed Hash Tables (DHTs)



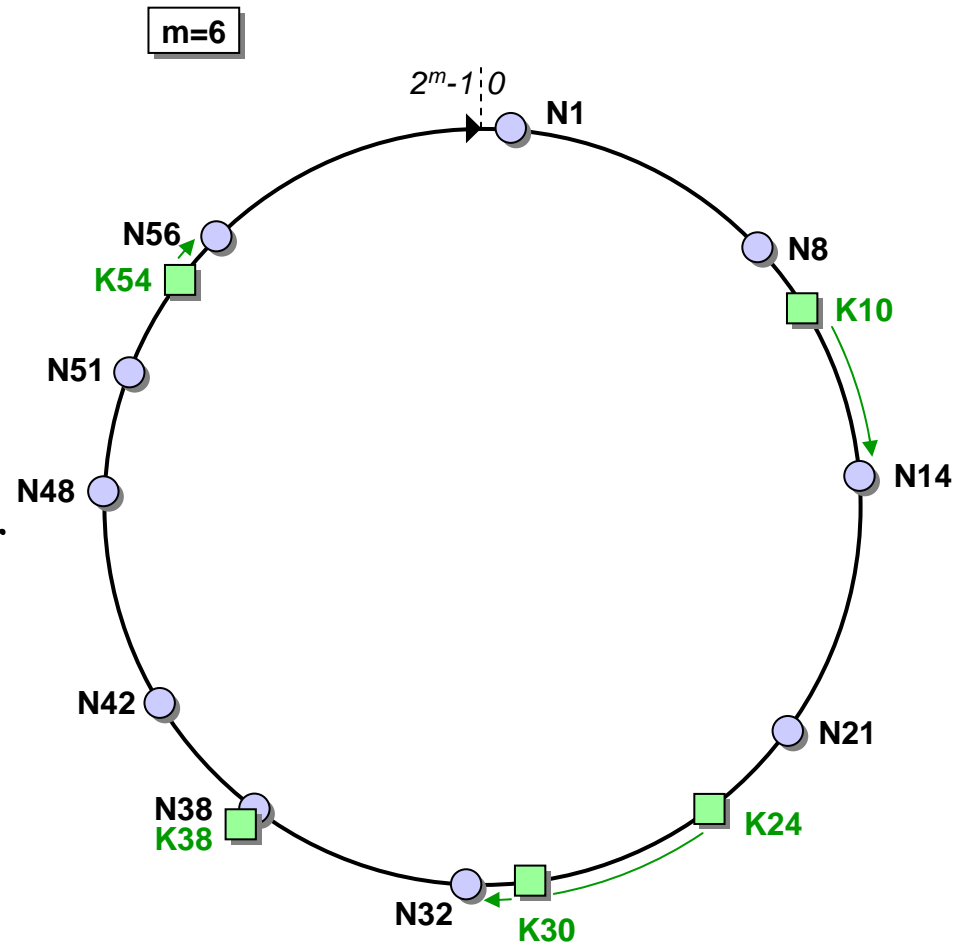
Distributed Hash Tables

- A hash table links data with keys
 - The key is hashed to find its associated *bucket* in the hash table
 - Each *bucket* expects to have $\#elems/\#buckets$ elements
- In a **Distributed Hash Table (DHT)**, physical nodes are the hash *buckets*
 - The key is hashed to find its responsible node
 - Data and load are balanced among nodes



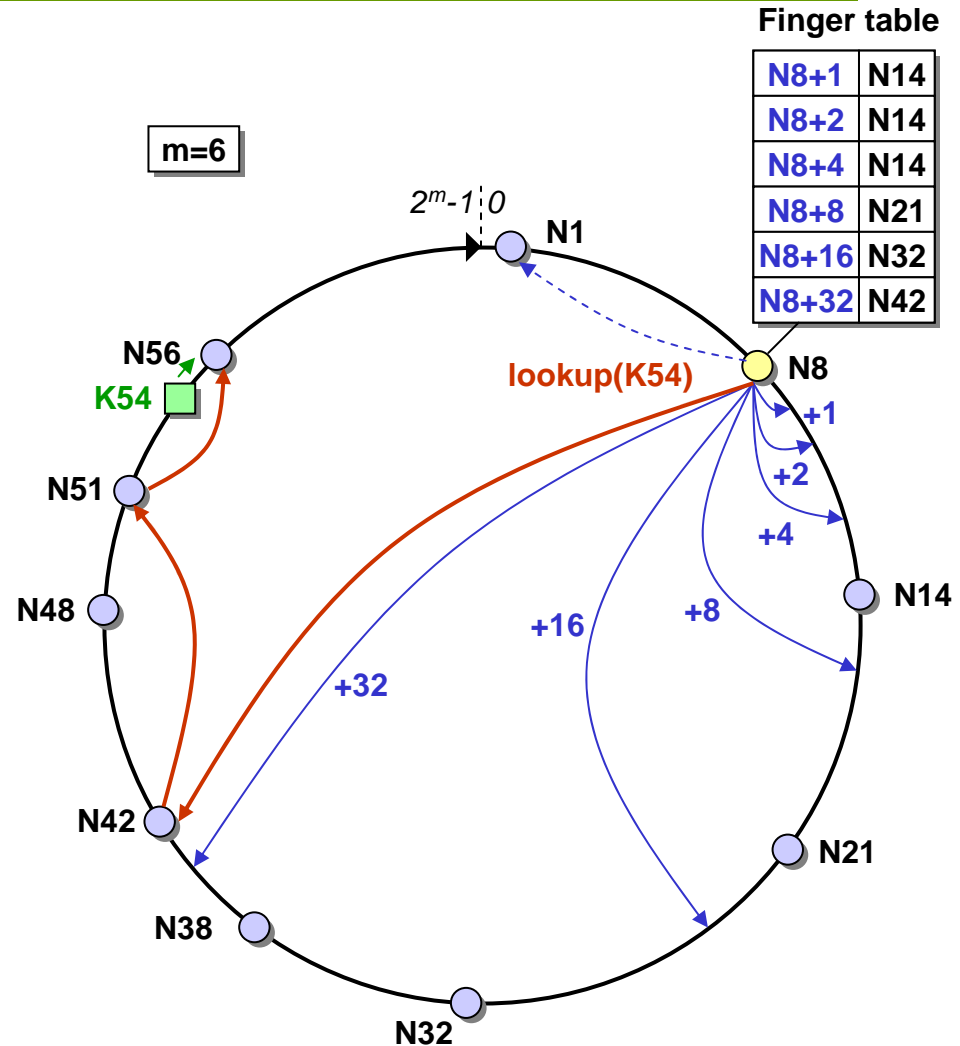
Case Study (MIT's Chord)

- Circular m -bit ID space for both keys and nodes
- Node ID = SHA-1(IP address)
- Key ID = SHA-1(key)
- A key is mapped to the first node whose ID is equal to or follows the key ID
 - Each node is responsible for $O(K/N)$ keys
 - $O(K/N)$ keys move when a node joins or leaves



Chord's State and lookup

- Each node knows m other nodes on the ring
 - Successors: finger i of n points to node at $n+2^i$ (or successor)
 - Predecessor (for ring management)
 - $O(\log N)$ state per node
- Lookup is achieved by following closest preceding fingers, then successor
 - $O(\log N)$ hops

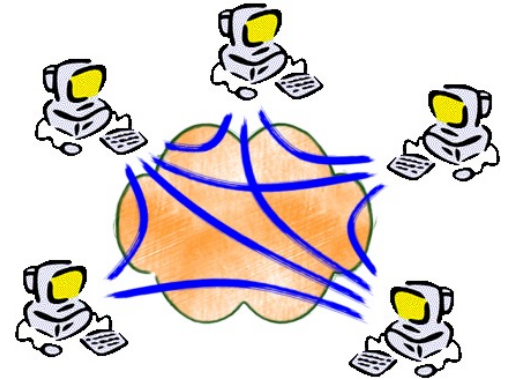


DHTs problems

- Routing state maintenance algorithms (overhead)
- They do not work well when lots of nodes join and leave frequently (churn)
- Their structured topology and routing algorithms make them frail and vulnerable to security attacks.
- Network locality problem: Nodes numerically-close are **not** topologically-close
- Commercial implementations are slowly arising (it is still a research problem)
- Security !

DHT applications

- File sharing [CFS, OceanStore, PAST, ...]
- Web cache [Squirrel, Coral]
- Censor-resistant stores [Eternity, FreeNet, ...]
- Application-layer multicast [Narada, ...]
- Event notification [Scribe]
- Streaming (SplitStream)
- Naming systems [ChordDNS, INS, ...]
- Query and indexing [Kademlia, ...]
- Communication primitives [I3, ...]
- Backup store [HiveNet]
- Web archive [Herodotus]
- Suggestions ?



Conclusions

- Structured topologies and DHTs provide interesting infrastructures for the Next Generation of P2P Killer apps
- Chord's commercial implementations are tough, but other DHTs can make it: Kademlia, Symphony, ..
- **But**, more research is needed to devise more robust, secure and efficient systems.

P2P Problems

- Tragedy of the commons
- Incentives !!! Tit-for-tat
- Heterogeneity
- Network proximity (hops !)
- Churn
- Security

Conclusions

- P2P killer application can emerge in the next years thanks to improvements in connectivity and computing power
- Success stories: Kazaa, Bittorrent, Skype, Groove
- But also local: MP2P, Filetopia
- Mobile P2P & Pervasive computing ??
- Google is hiring P2P researchers !!

- New Ideas ??