Διομότιμα Δίκτυα

Peer-to-Peer Networks

Peer-to-Peer Systems, R. Rodrigues and P. Druschel, CACM, October 2010

Slides from: Isabelle Demeure, ENST; Demetris Zeinalipour, OUC; Evangelos Markatos, ICS/FORTH; Seif Haridi, KTH

Some Historical Notes

Introduction

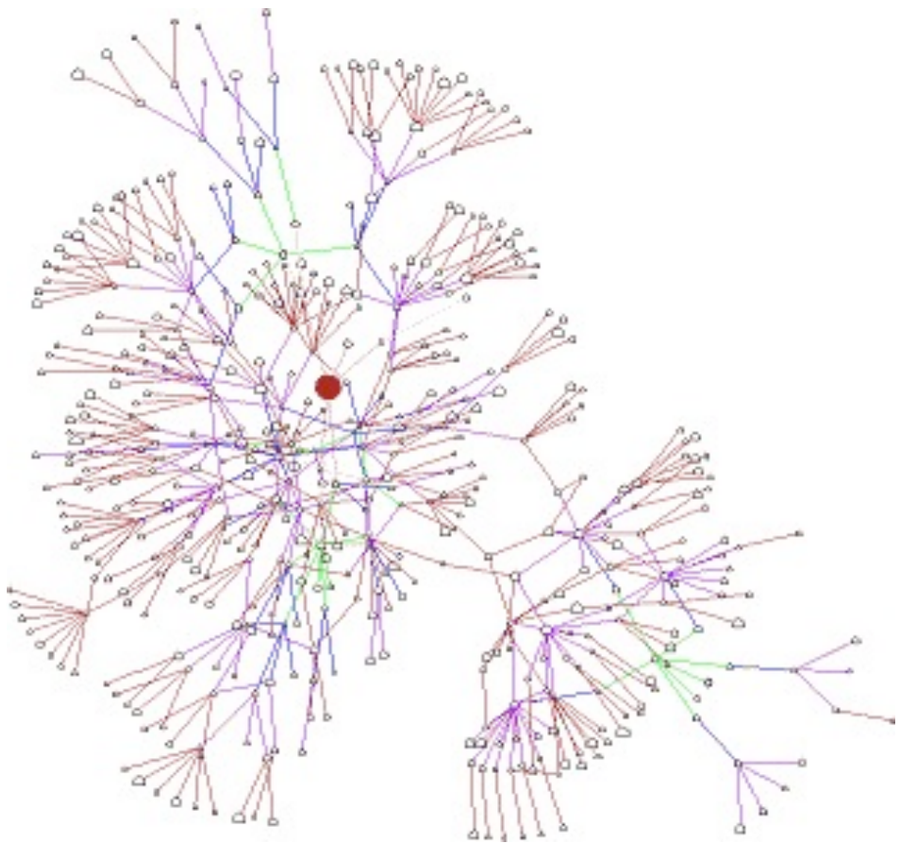
- Peer-to-peer networking:
  - a set of technologies that enable the direct exchange of services or data between computers
- Started with Napster and MP3 file exchange:
  - raised legal copyright issues
  - there's something attractive about the defiance or avoidance of authority
- Interest in P2P ranges from
  - enthusiasm, through hype,
  - to disbelief in its potential.
- Currently (2006), 60% to 89% of all Internet traffic is due to p2p traffic.
  - A marketing argument ?
  - Still interesting architectural and technical issues behind ...

Peer-to-Peer Systems: History

- 1999
  - Shawn Fanning invented Napster a song-sharing program
  - Napster users shared the songs stored on their computers
  - They could search for other users in Napster's central servers

- 2001
  - Napster was facing law suits and was shut down/restructured
    - It closed down its central servers

Peer-to-Peer Systems: History



- 2001
- Gnutella came along
  - Distributed search mechanisms
  - Users are organized in a graph of neighbors
  - Nodes ask their neighbors for song
    - Neighbors ask their neighbors and so on
  - More difficult to shut down
    - Nodes enter/leave all the time

History (II)

- 2001
  - KaZaA came along
    - The first major security incident:

      - Installs Malware on client computers
      - Advertisements, spyware, hijackers
  - DC++ started

    Partially Decentralized

    Based on "hubs" of common Interests

    Users must join a hub in order to download songs from the clients of the hub.

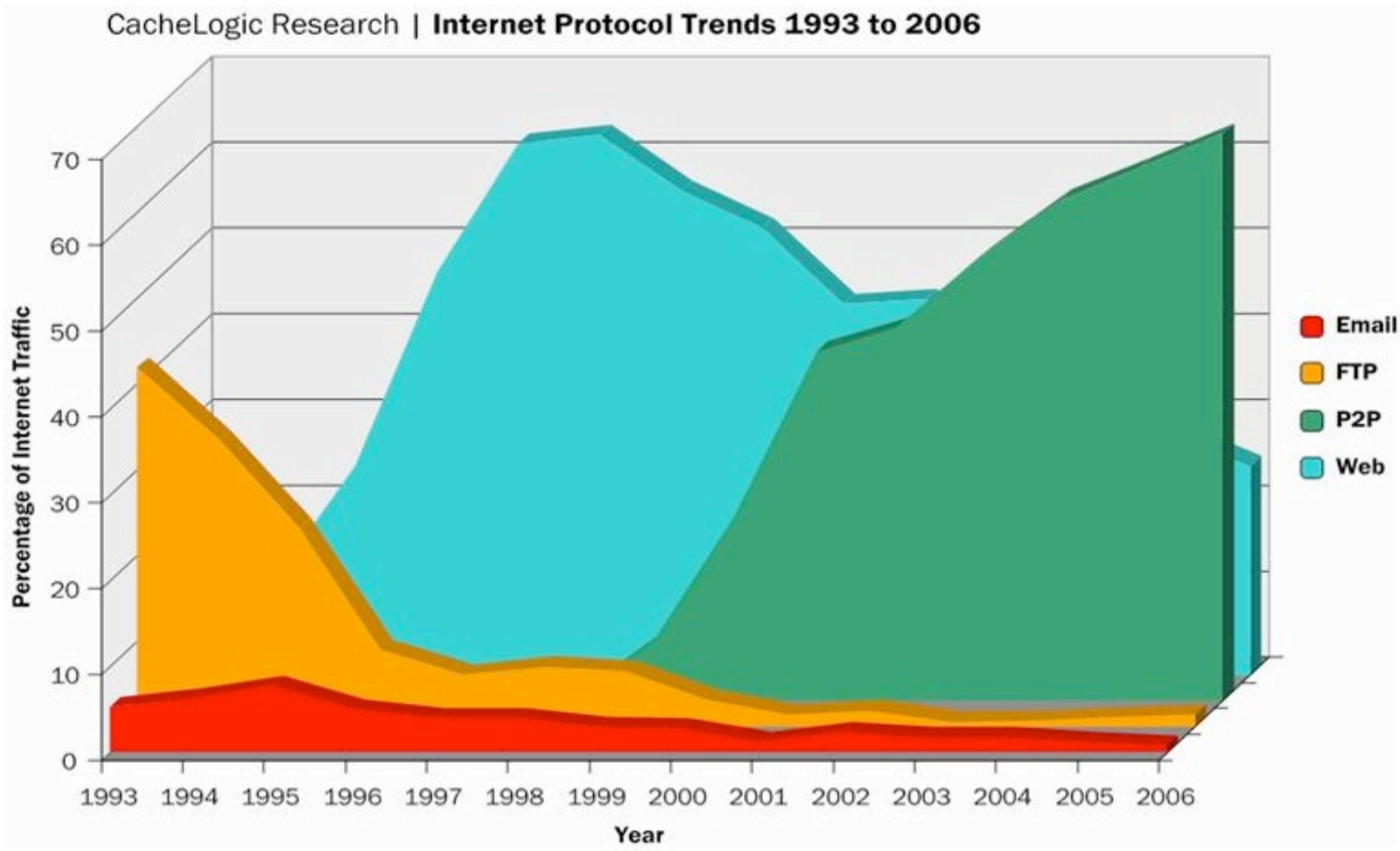
    Hubs may have "acceptance rules"
      - Minimum size of data shared
      - Accessible only to clients of particular ISPs, countries, etc.

History (III)

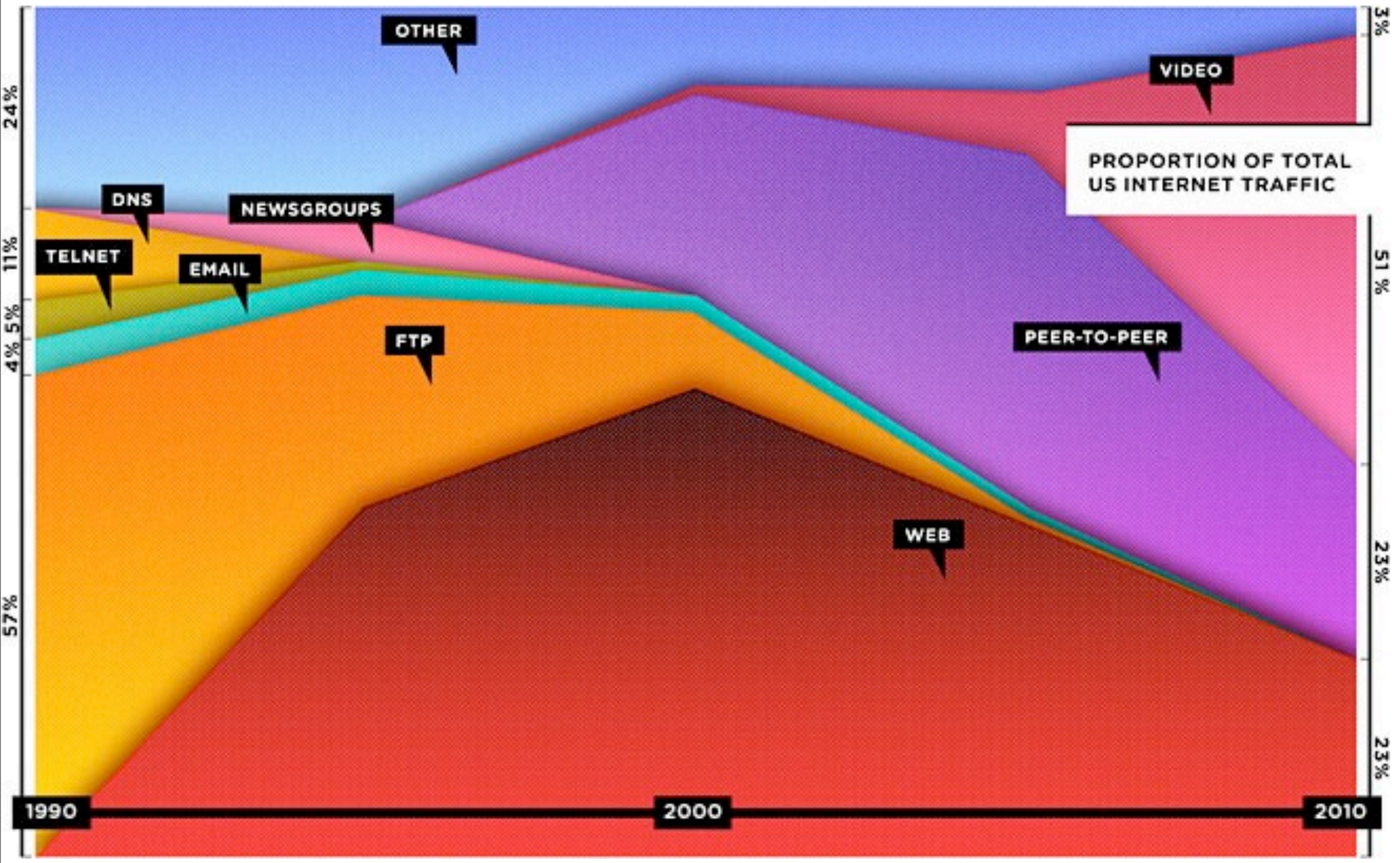
- 2002
  - eMule eDonkey
    - Based on Servers
- 2003
  - Bittorrent
    - A computer (called tracker) describes who has the file
    - Tit-for-tat approach

BitTorrent™

Today (2006)



- P2P is the **single largest application on the Internet**


OTHER
VIDEO
3%
PROPORTION OF TOTAL
US INTERNET TRAFFIC
DNS
NEWSGROUPS
TELNET
EMAIL
FTP
PEER-TO-PEER
WEB
24%
11%
4%
5%
57%
51 %
23%
23%
1990
2000
2010

P2P evolution

- P2P systems is a unique specie in the IT family with deep-rooted properties
- P2P systems evolved under a lot of pressure
  - Constant chase of DRM companies
  - Constant threat of lawsuits
  - Request for anonymity from users

P2P Systems evolution

- So, they evolved
  - They are agile – highly dynamic
    - They do not have constant URLs
    - Clients come and go all the time
  - They provide anonymity
    - Difficult to trace who is who
    - Clients do not have a domain name
    - Clients may frequently change
      - IP address and IP ports
- It seems that they are the perfect place to hide illegal activities

Child Abuse

United States General Accounting Office

GAO

Testimony

Before the Subcommittee on Commerce, Trade, and Consumer Protection, Committee on Energy and Commerce, House of Representatives

For Release on Delivery
Expected at 10:00 a.m. EDT
Thursday, May 6, 2004

FILE SHARING PROGRAMS

Users of Peer-to-Peer Networks Can Readily Access Child Pornography

Statement of Linda D. Koontz
Director, Information Management Issues

G A O

GAO-04-757T

- P2P networks are currently being used to circulate child abuse pictures
- As much as 44% of the images are child abuse
- Even when searching for popular singers/ actors one may stumble upon child abuse pictures.

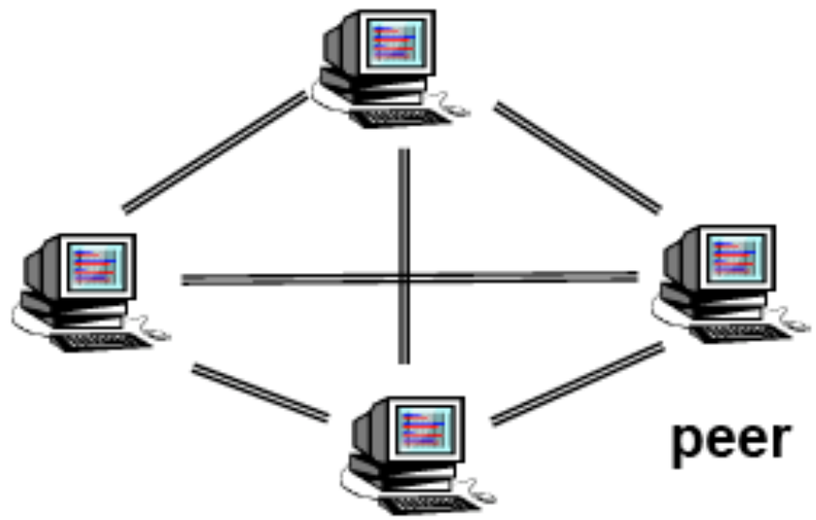
Definitions

first, some terminology

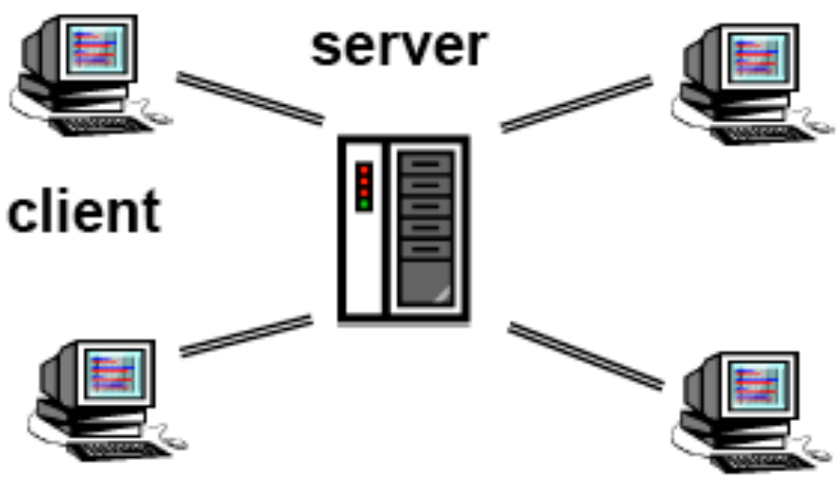
- Ad-hoc communication:
  - Enables communication to take place without any preexisting infrastructure in place, except for the communicating computers.
  - The ad hoc network takes care of communication, naming, and security.
  - P2P systems can be used on top of an ad-hoc communication infrastructure.
- Overlay network or virtual network
  - A network built on top of one or more existing networks
  - Adds an additional layer of indirection/virtualization
  - Changes properties in one or more areas of underlying network

P2P - Informal definition

- "Peer-to-peer" (P2P) refers to a class of systems and applications that employ distributed resources to perform a function in a decentralized manner



- Generally opposed to the client/server architecture.



What is P2P?

- "The sharing of computer resources and services by direct exchange between systems" [p2pwg, 2001].
  - enables peers to share their resources (information, processing, presence, etc.) with at most a limited interaction with a centralized server.
  - equivalent to having all entities being client and servers (servents) for the same purpose.

What is P2P?

- A distributed system with the following properties:
  - **High degree of decentralization**: few if any dedicated nodes with centralized state
  - **Self-organization**: once a node enters the system, little or no manual configuration needed to maintain the system
  - **Multiple administrative domains**

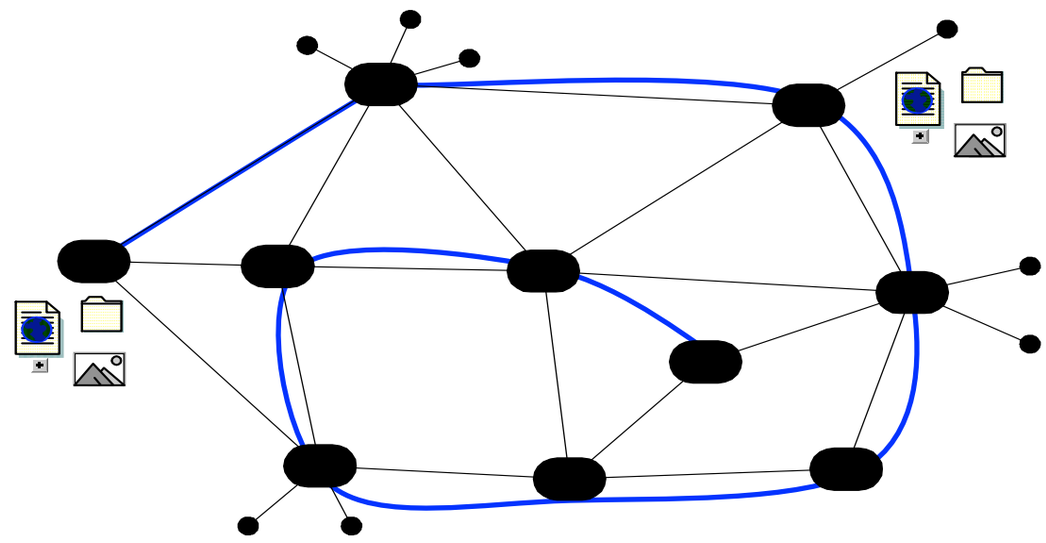
Some definitions of P2P

- "The sharing of computer resources and services by direct exchange between systems" [p2pwg, 2001].
- "The use of devices on the internet periphery in a non-client capacity" [Veytsel, 2001].
- P2P systems follow three key requirements [Graham 2001]:
  - they have an operational computer of server quality
  - they have an addressing system independent of DNS
  - and they are able to cope with variable connectivity.
- "P2P is a class of applications that takes advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers" [Shirky 2001].

P2P Systems and Overlay Networks

- Τα διομότιμα συστήματα είναι δομημένα σαν υπερκείμενα δίκτυα (overlay networks)
  - –Οι διομότιμοι κόμβοι εγκαθιδρύουν TCP ή UDP συνόδους μεταξύ τους, δημιουργώντας έναν «εικονικό» γράφο πάνω από τον γράφο του διαδικτύου.
  - –Η τοπολογία του εικονικού γράφου είναι ανεξάρτητη από την τοπολογία του διαδικτύου
  - –Ο εικονικός γράφος δεν αντιπροσωπεύει τον γράφο των δικτυακών μονοπατιών ανάμεσα στους κόμβους του P2P δικτύου

The physical topology ——
The virtual P2P topology ——



P2P vs Client-Server

Client/Server Computing

- What is it?
  - 2 asymmetrical roles: client and server
  - Operate in a request-response (or service-delivery) mode
- Terminology
  - Client:an entity that initiates requests but is not able to serve them.
  - Server: an entity that serves requests from other entities but does not initiate requests
  - Client-Server model: represents the execution of entities with the roles of clients and servers.
- Examples:
  - web, FTP, telnet, email use the client/server model
- Characteristics of C/S computing:
  - Most of the computing burden is on the servers
  - Clients may give away unused resources

Χαρακτηριστικά συστημάτων Πελάτη-Εξυπηρετητή (Π/Ε)

- Πλεονεκτήματα
  - Κεντρικοποιημένη διαχείριση δεδομένων
  - Ασφάλεια δεδομένων
  - Αυτά τα χαρακτηριστικά είναι ιδανικά για πολλές Διαδικτυακές Εφαρμογές (Συστήματα Τραπεζών, Ηλεκτρονικού Ταχυδρομείου, κτλ.)
- Μειονεκτήματα
  - Χρειάζονται ακριβές υπολογιστικές υποδομές (π.χ., το Youtube.com αναλώνει 25TB – 250TB ανά μήνα και πληρώνει $5M/μήνα ή $170K/μέρα για το bandwidth)
  - Σε περίπτωση βλάβης του εξυπηρετητή χάνουμε την υπηρεσία (single point of failure).
  - Χρειάζονται συνεχή διαχείριση (administration).
  - Μπορούν να λογοκριθούν (censorship) και να ελεγχθούν (από καθεστώτα, κτλ) π.χ., τα αποτελέσματα αναζήτησης μηχανών αναζήτησης ελέγχονται από το κράτος στην Κίνα.
- Τα P2P Systems προσπαθούν να ξεπεράσουν αυτά τα προβλήματα.

Λόγοι Ανάπτυξης των P2P

- Ο μεγάλος αριθμός PCs με πολύ ψηλή υπολογιστική ισχύ διαθέσιμα στα άκρα (edges) του Internet.
- Επίσης, υπάρχει πολύ ψηλότερο bandwidth διαθέσιμο στα άκρα του Internet (ADSL, Satellite, Cable, Ethernet LANs, etc.)
- Επομένως, γίνεται εφικτό να αξιοποιήσουμε το Storage, Cycles, Content και να έχουμε αλληλεπίδραση Η/Υ στα άκρα (edges) του Internet, χωρίς την χρήση κεντρικών servers.
  - Σημειώστε ότι το άλλα Internet Services από το 1980- π.χ. DNS, BGP, Usenet etc, ορίζουν και αυτά ένα P2P μοντέλο ανταλλαγής πληροφοριών

Available resources all over the Net

- Assume 10 million 2 GHz client machines are connected to the Internet.
- If each has 100Mbytes of unused storage space
  - Then there is 10 Petabytes (10^{15}) of unused storage space
- If each has 1Kbit/s of unused bandwith
  - Then there is 10 billion bit/s (1.25 Gbyte/s) of available bandwith
- If each has 10% of unused processing power
  - Then there is 2 million GHz of unused processing power.
  - Explains interest in P2P computing that makes use of unused resources

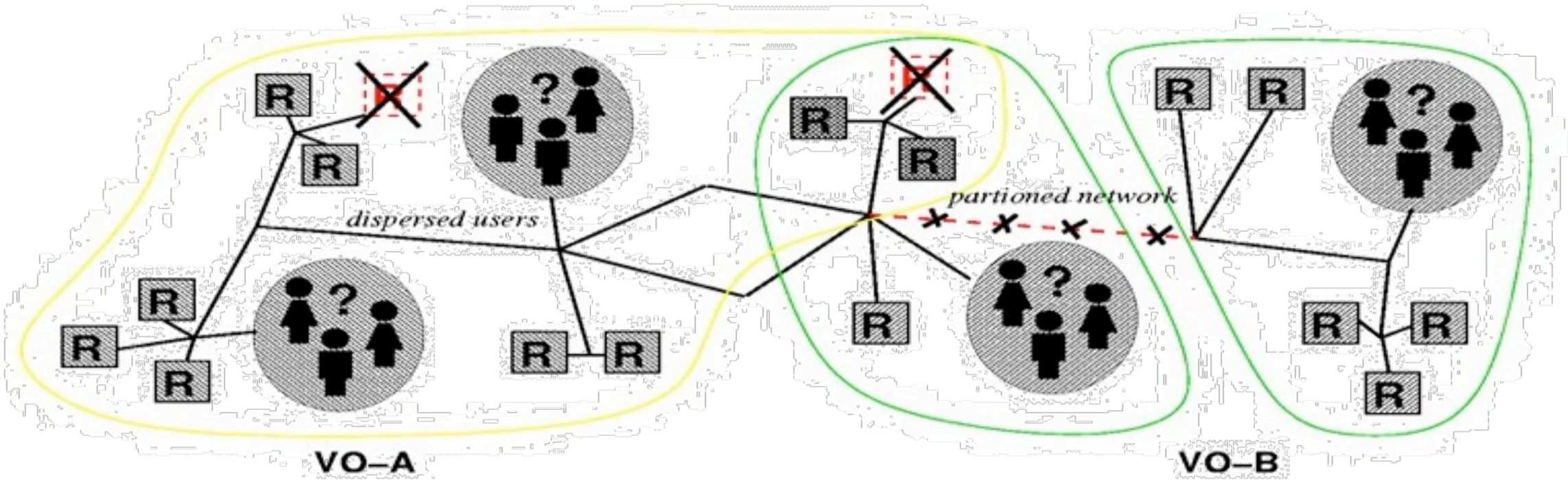
P2P vs Grids

P2P vs Grids

- Grid computing:
  - "grid problem": “Coordinated resource sharing and problem solving in large, multi-institutional virtual organization.” [Foster 1999].
  - The grid thus refers to an infrastructure that enables the integrated, collaborative use of high-end computers, networks, databases, and scientific instruments owned and managed by multiple organizations.

A Unifying Concept: The Grid

- "Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations"



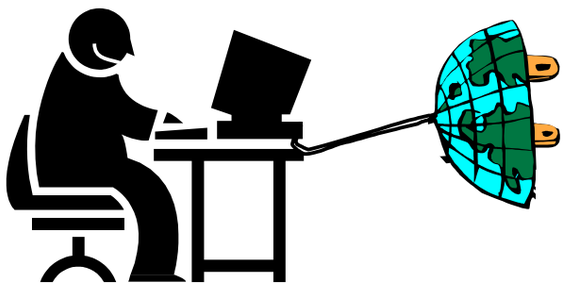
Enable integration of distributed resources
Using general-purpose protocols & infrastructure
To achieve better-than-best-effort service

Grid: the Five big Ideas

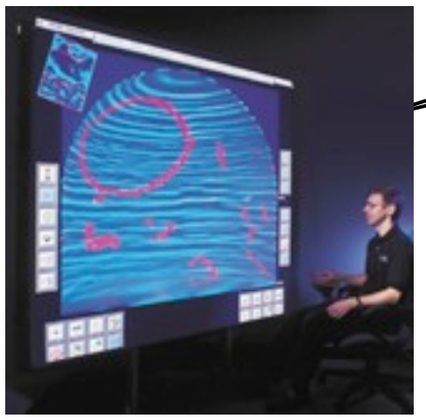
- The most important is the sharing of resources on a global scale; the very essence of the Grid.
- Security is a critical aspect of the Grid; establishing trust.
- The Grid really starts to pay off when it can balance the load on the resources, so that computers everywhere are used more efficiently
- Distance no longer matters.
- Use open standards to make sure that R&D worldwide can contribute.

The Grid Metaphor

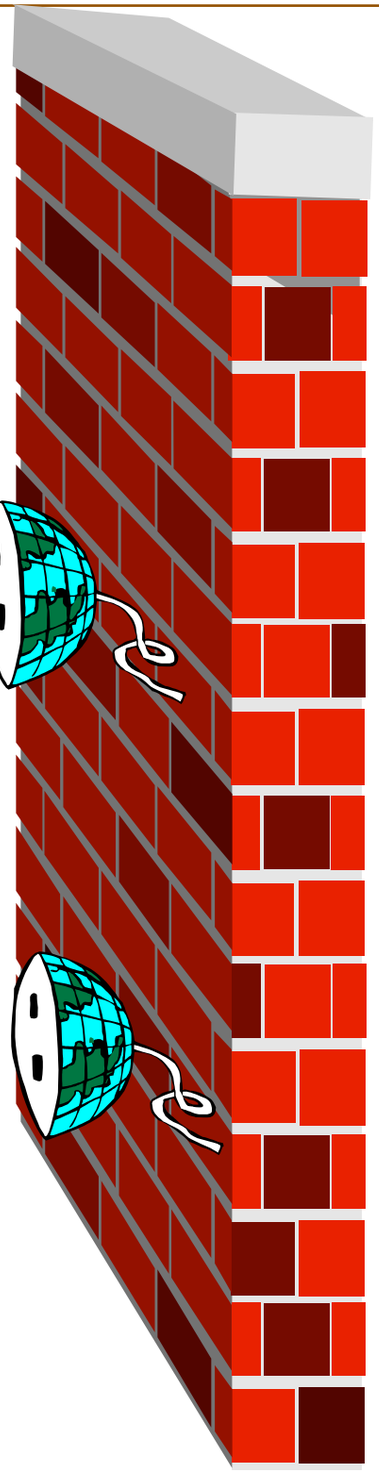
Mobile Access



Desktop

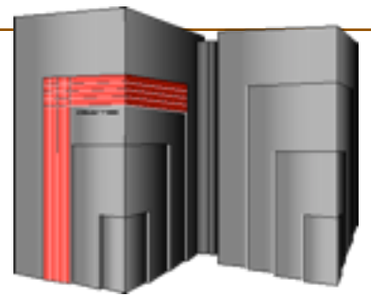


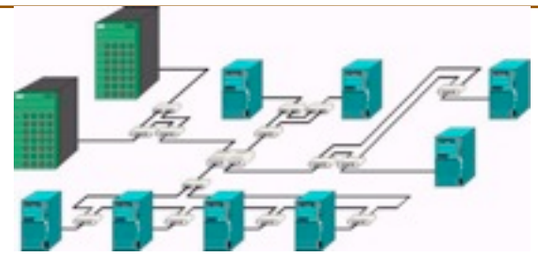
Visualization



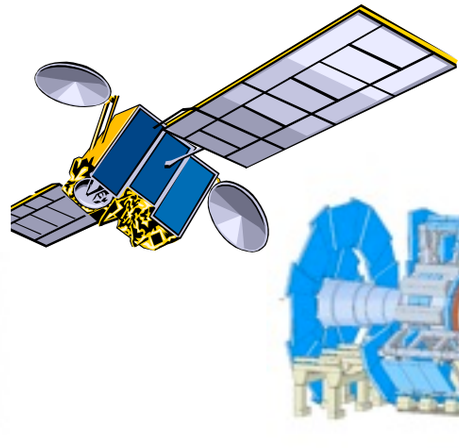
G
R
I
D

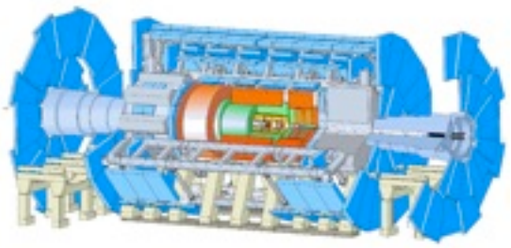
M
I
D
D
L
E
W
A
R
E



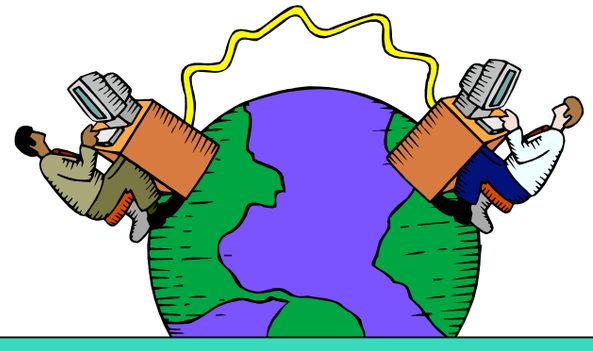


Supercomputer, PC-Cluster





Data-storage, Sensors, Experiments



Internet, networks

Hoffmann, Reinefeld, Putzer

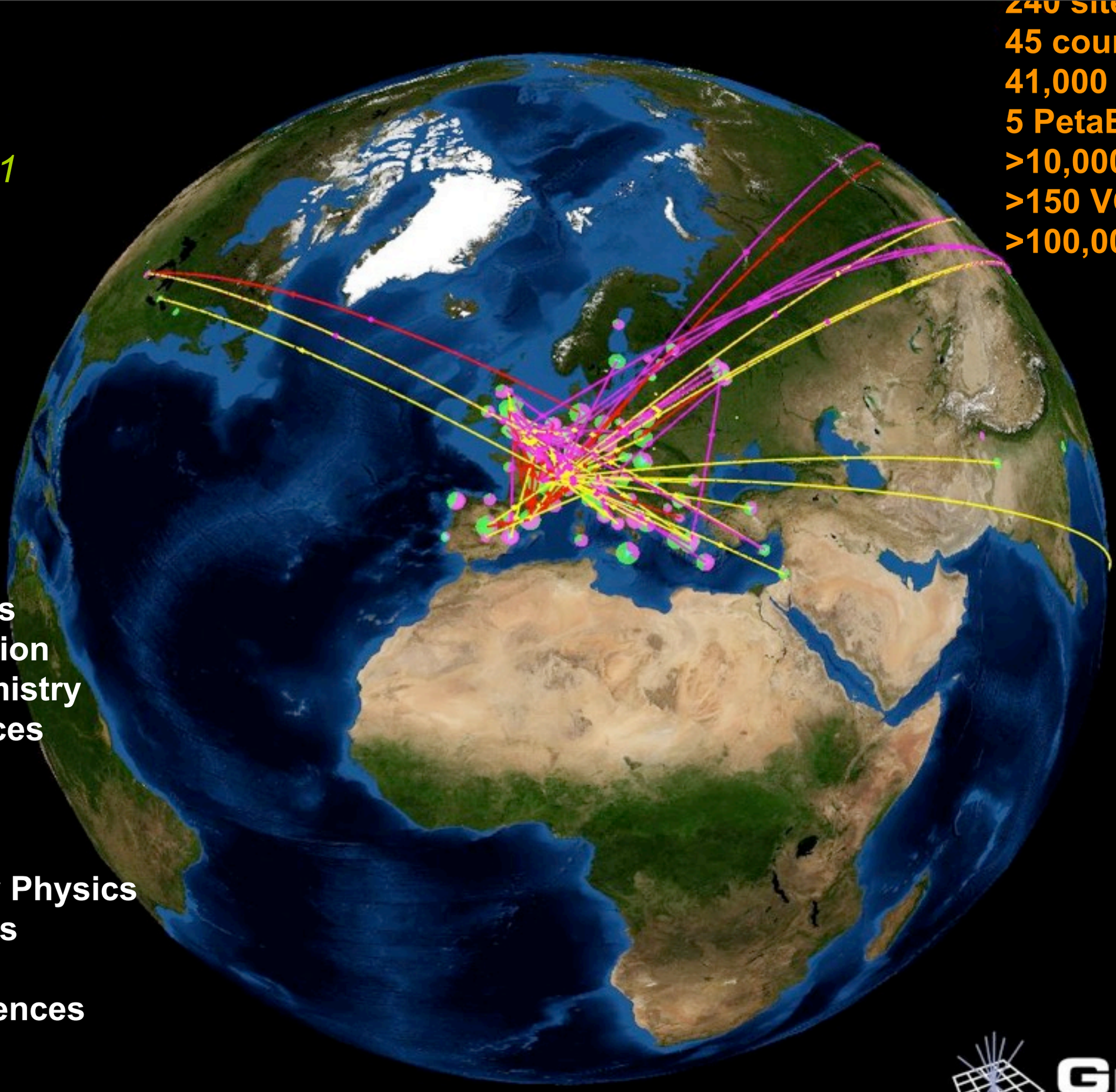
What Can It Do?
Computer vs Data vs Community

- Computer-centric problems:
  - I need teraflops, lots of them!
  - The Grid combines large computational resources
- Data-centric problems:
  - The Grid is used to collect, store and analyze data maintained in geographically distributed repositories, digital libraries, and databases.
- Community-centric problems:
  - Collaborative applications -- enable and enhance human-to-human interactions.
  - Provide a “virtual shared space”

Enabling Grids for E-sciencE

2004-2011

240 sites
45 countries
41,000 CPUs
5 PetaBytes
>10,000 users
>150 VOs
>100,000 jobs/



Archeology
Astronomy
Astrophysics
Civil Protection
Comp. Chemistry
Earth Sciences
Finance
Fusion
Geophysics
High Energy Physics
Life Sciences
Multimedia
Material Sciences
…

GridPP
UK Computing for Particle Physics

P2P vs Grids

- Both are concerned with the same general problem
  - Resource sharing within virtual communities (not the same communities)
  - Both take the same general approach
  - Creation of overlays that need not correspond in structure to underlying organizational structures
- Grid applications often involve large amounts of data and/or computing.
- The problems tackled by current P2P applications do not require large resource commitments from the peer nodes
- Each has made genuine technical advances, but in complementary directions
  - "Grid addresses infrastructure but not yet failure"
  - "P2P addresses failure but not yet infrastructure"
  - Complementary strengths and weaknesses => room for collaboration (Foster)

Properties and Key Characteristics

Why are P2P interesting?

- Interesting characteristics:
  - –Low barrier to deployment: no upfront costs
  - –Organic growth
  - –Resilience to faults and attacks
  - –Abundance and diversity of resources
- Challenges:
  - –Manageability
  - –Security
  - –Law enforcement

P2P Advantages

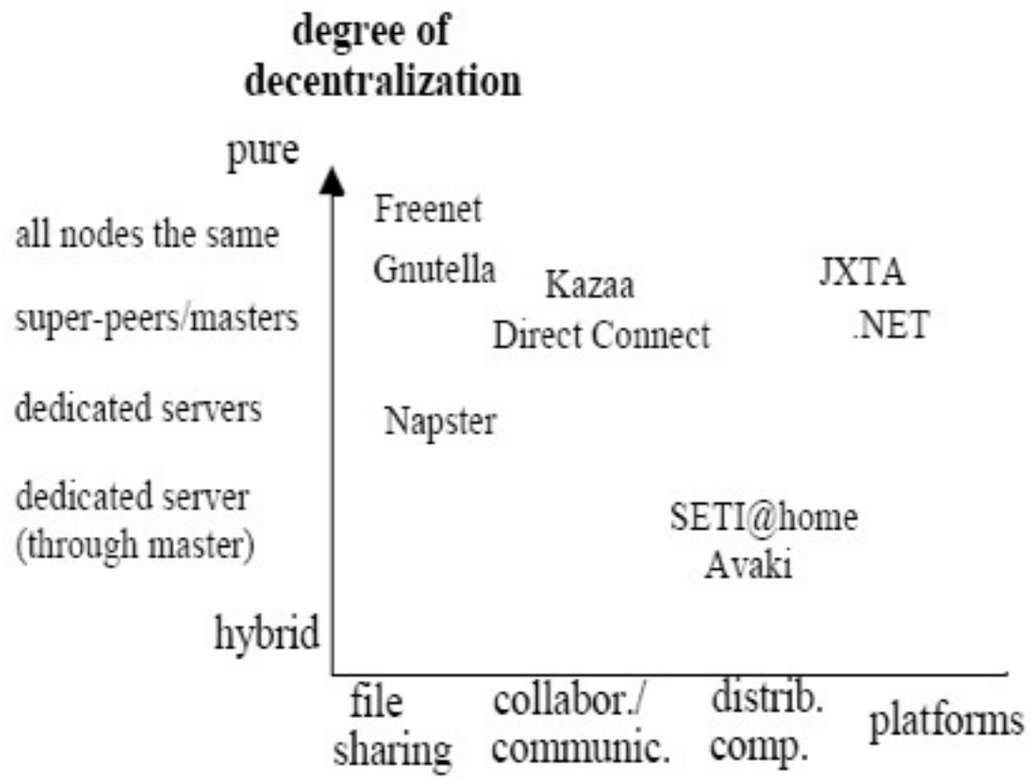
- Edge-Computing
  - Αξιοποίηση αχρησιμοποίητου bandwidth, storage, processing power στα άκρα (edge) του Internet
- Scalability
  - Δεν υπάρχει συμφόρηση (bottleneck) σε κάποιο κεντρικοποιημένο κόμβο. Επομένως τα συστήματα αυτά μπορούν να μεγαλώνουν “απεριόριστα”.
  - Oct 12th Gnutella (Limewire.com): 2,219,539 κόμβοι
- Reliability (Αξιοπιστία)
  - No single point of failure, Γεωγραφική Κατανομή Περιεχομένου (CDNs)
- Ease of administration
  - Οι κόμβοι οργανώνονται μεταξύ τους αυτόματα (self-organization).
  - Αυτόματα επίσης γίνεται το replication και το load balancing καθώς τέτοια συστήματα παρέχουν fault tolerance.
- Anonymity – Privacy
  - …κάτι το οποίο δεν είναι εύκολο σε ένα κεντρικοποιημένο σύστημα

P2P key features

- Οι χρήστες του συστήματος έχουν διπλό ρόλο: δηλ. είναι Πελάτες και Εξυπηρετητές την ίδια στιγμή.
- Καθώς ο χρήστης Α εξυπηρετείται από τον χρήστη Β, κάποιος χρήστης Γ μπορεί να εξυπηρετείται από τον Α.
- Δεν υπάρχει κεντρικοποιημένη διαχείριση…το οποίο δημιουργεί ένα αίσθημα ελευθερίας.
- …Παράλληλα όμως δημιουργεί και πολλά προβλήματα (όπως αυτό της παράνομης ανταλλαγής τραγουδιών)

Key characteristics and concerns

- Decentralization:
  - –free of bottlenecks
  - –lack of single point of failure
  - –ownership and control of processing, storage space, bandwidth, and content shared by many
  - –Difficulty: finding the network, discovery of resources, insertion of new nodes



Key characteristics and concerns

- Massive scalability
  - –no centralized resources to exhaust
  - –limited by factors such as:
    - amount of centralized operations
    - amount of *state* to be maintained
    - inherent parallelism an application exhibits
    - programming model
- Organization can become the scarce resource as the P2P networks grow very large
  - –need algorithms to manage peer connections, communicating requests and responses
  - –these algorithms must support self-organization and fault resilience

Key characteristics and concerns

- Resource ownership is shared
  - The cost of obtaining and maintaining each peer is born by its owner
  - Shared ownership reduces the cost of maintaining the system and owning the systems.
  - Responsibility for creating, publishing and distributing content is shared
- Ad-hoc connectivity
  - Peers join and leave the network at any time
  - Peers are not available at all times. Hence, P2P systems need to be aware of this ad-hoc nature.
  - P2P systems need to tolerate disconnection and ad-hoc additions to groups of peers.

Key characteristics and concerns

- Security:
  - there are no established trust relationships between collaborating peers
  - need algorithms to manage trust, to isolate peers, to establish reputation etc
- Fault Resilience
  - P2P systems should deal with:
    - Disconnections/unreachability
    - Partitions
    - Node failures
  - Solutions:
    - Replication of crucial resources
    - Relays
    - Queuing messages

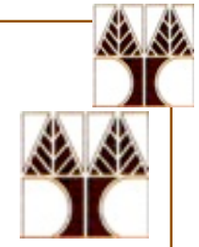
Anonymity

- Goals:
  - To allow people to use systems without concern for legal or other ramifications.
  - To guarantee that censorship of digital content is not possible.
- Types of anonymity: server, receiver, mutual

Self-Organization

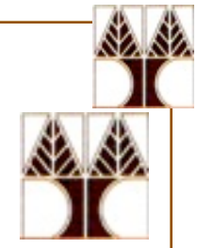
- P2P systems can scale unpredictably in terms of the number of systems, number of users, and the load.
- Adaptation is required to handle the changes caused by peers connecting and disconnecting from the P2P systems.
- For example in the FastTrack system, more powerful computers automatically become SuperNodes and act as search hubs. Any client can become a SuperNode if it meets processing and networking criteria (bandwidth and latency).

Performance



- P2P systems aim to improve performance by aggregating distributed storage capacity.
- Performance is influenced by:
  - Processing, storage and networking
- Approaches to optimize performance:
  - Replication, caching, intelligent routing

Transparency and Usability

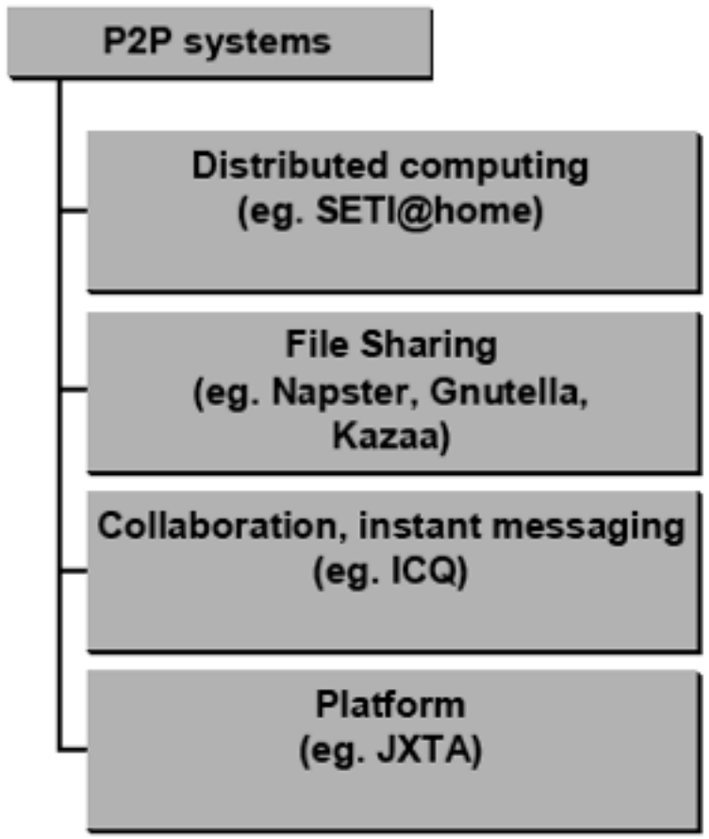


- End-to-end transparency: TCP/IP
- Naming transparency
- Network and device transparency
- Automatic and transparent authentication of users

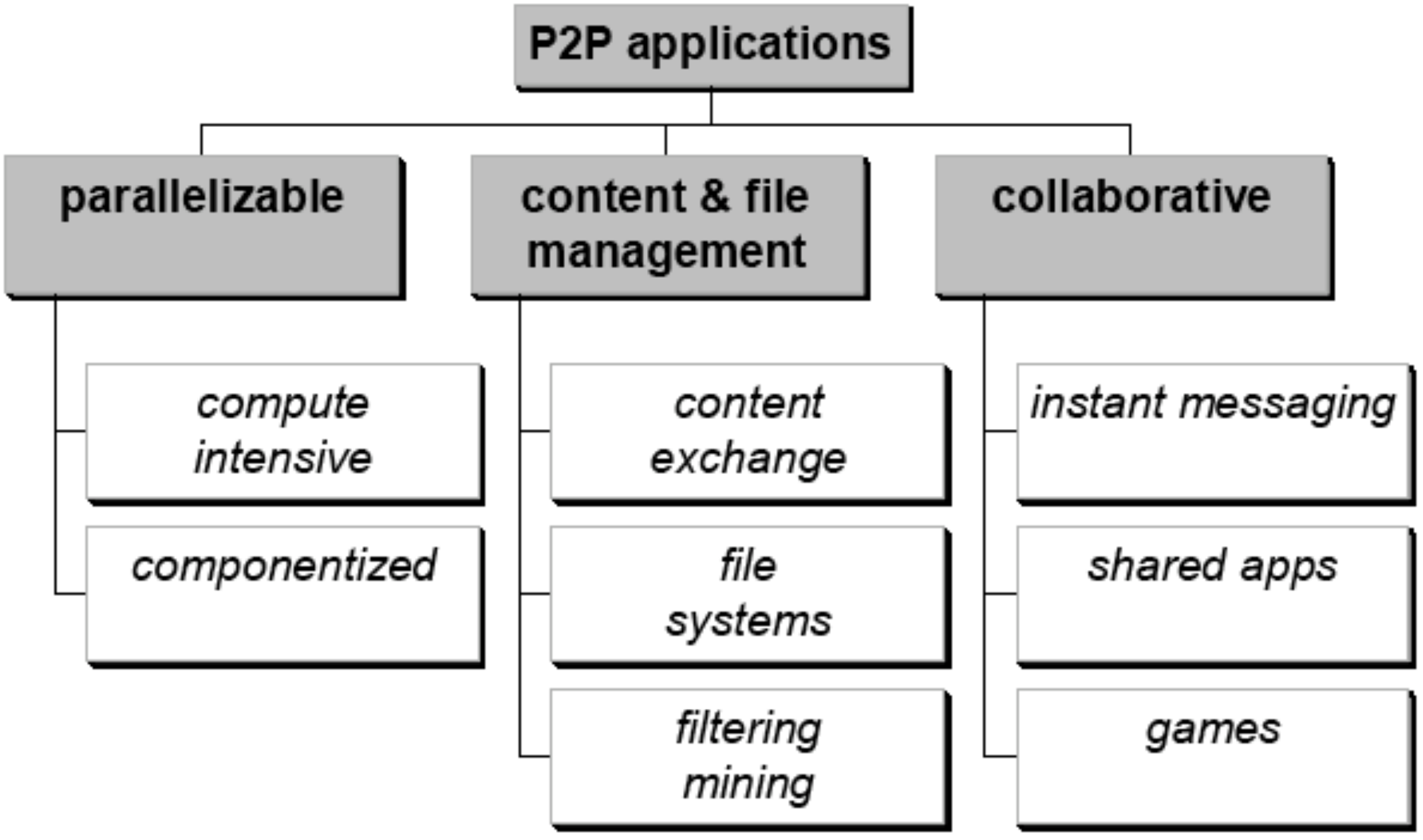
Applications

P2P Applications

- Ανταλλαγή Αρχείων (Napster, Gnutella, Bittorent, …)
- Διαδικτυακή Τηλεφωνία (Skype)
- Διαδικτυακά Παιχνίδια (Playstation Online Gaming)
- Πάταξη του Spam (SpamNet)
- Instant Messaging (IRC, MSN & Yahoo Msgers)
- Content Distribution Networks (CorelCDN)
- P2P Web Caching (Squirrel)
- Application-Level Multicast (Narada)



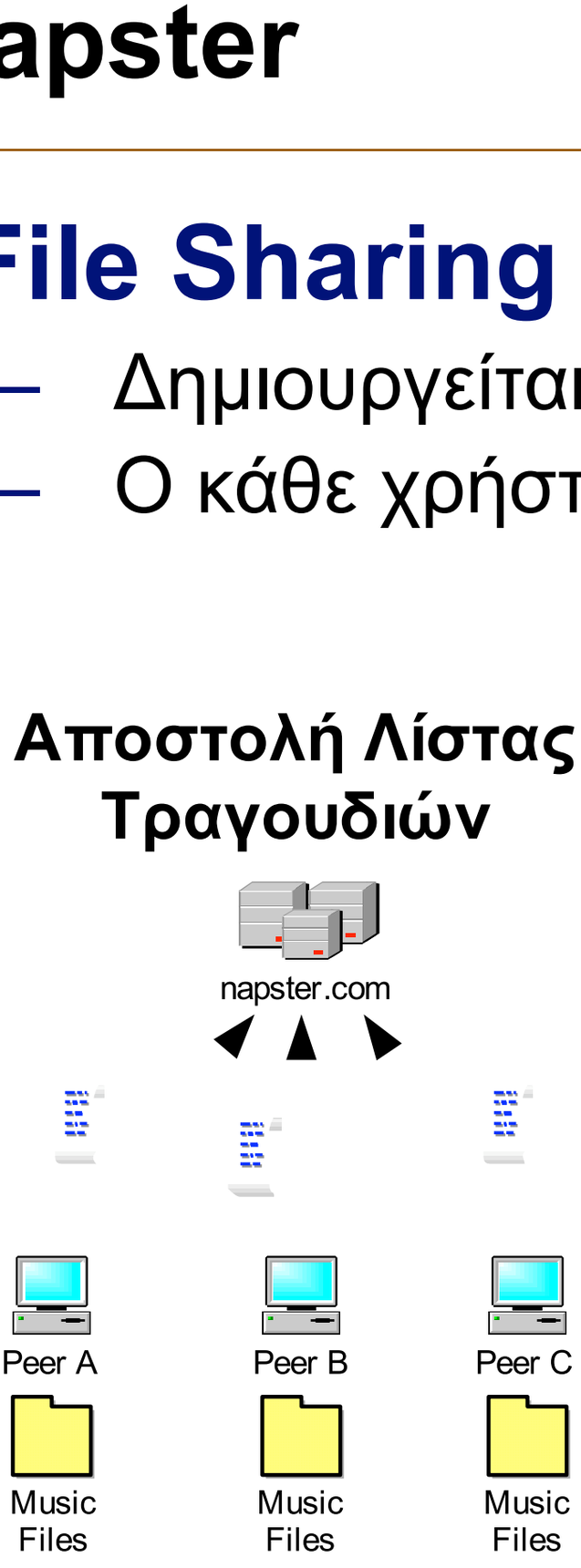
P2P Applications



Napster

- **File Sharing**
  - Δημιουργείται το 1999 από ένα 18-χρονο φοιτητή.
  - Ο κάθε χρήστης εκτελεί τις ακόλουθες τρεις διαδικασίες

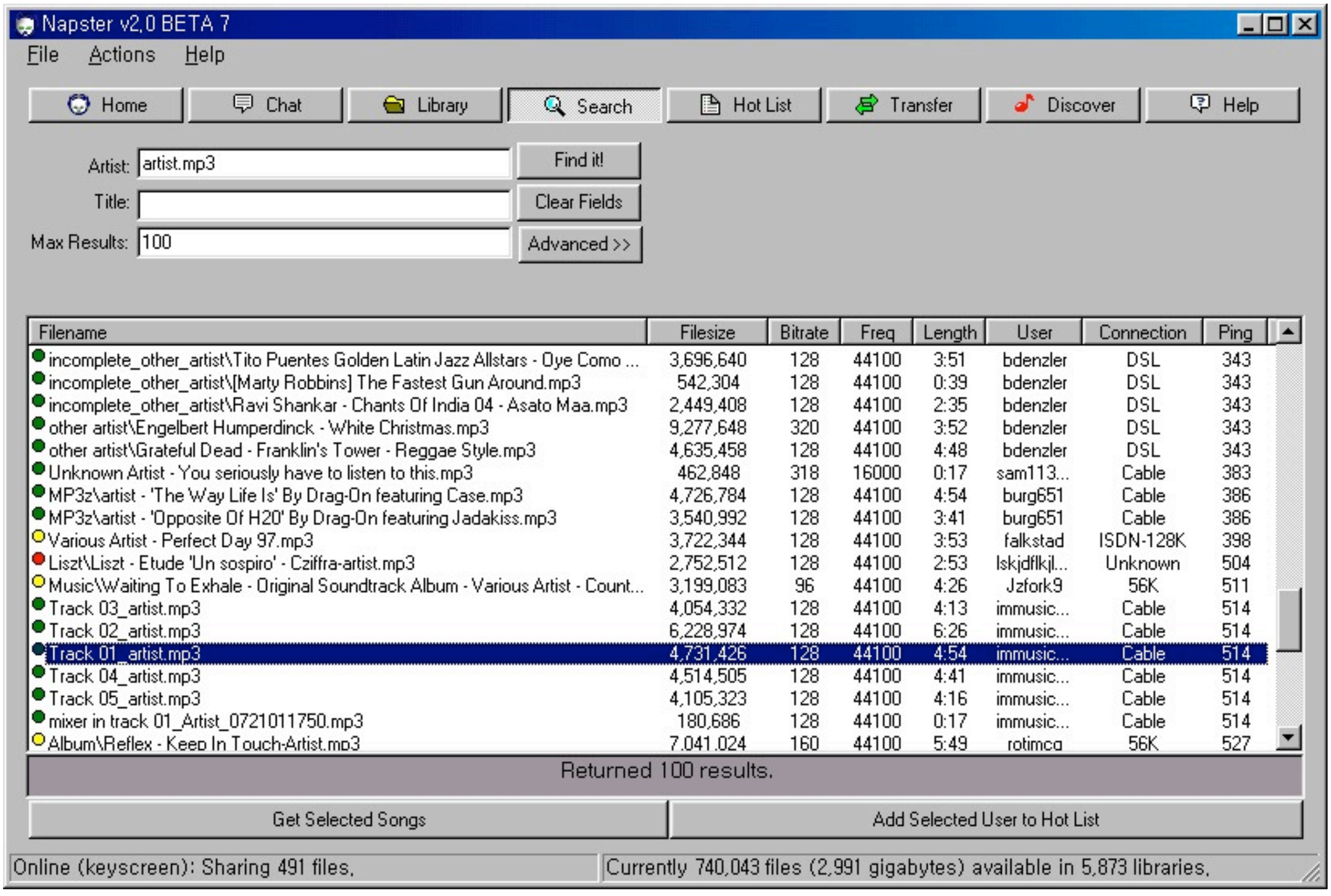
Αποστολή Λίστας Τραγουδιών



Αναζήτηση Τραγουδιών

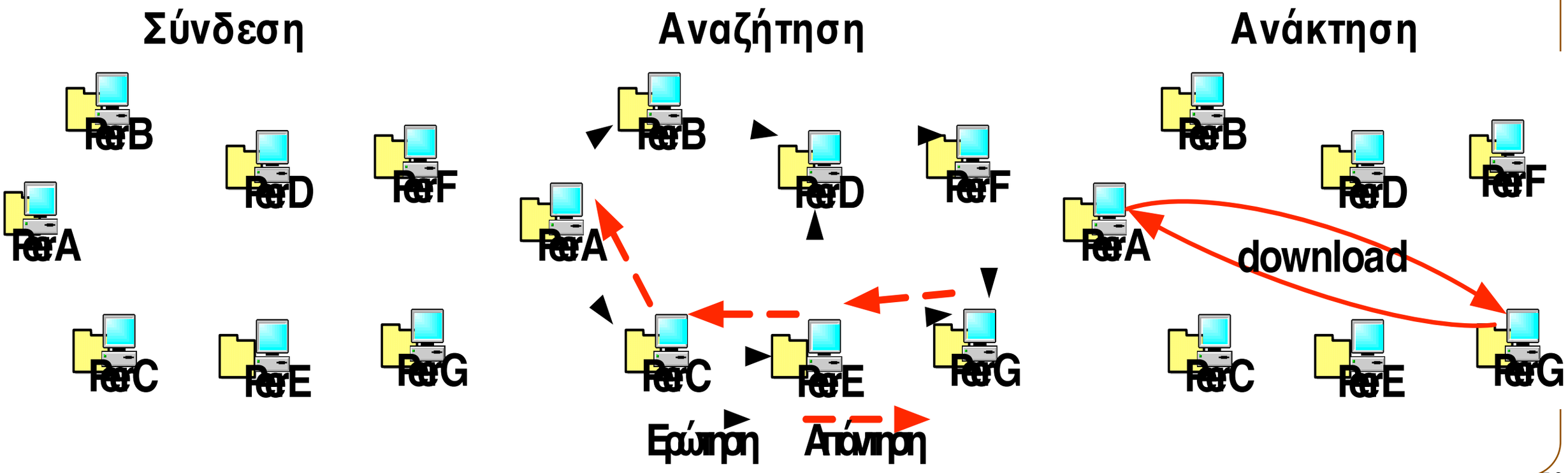
Ανάκτηση Τραγουδιών

Παράθυρο Αναζήτησης Napster

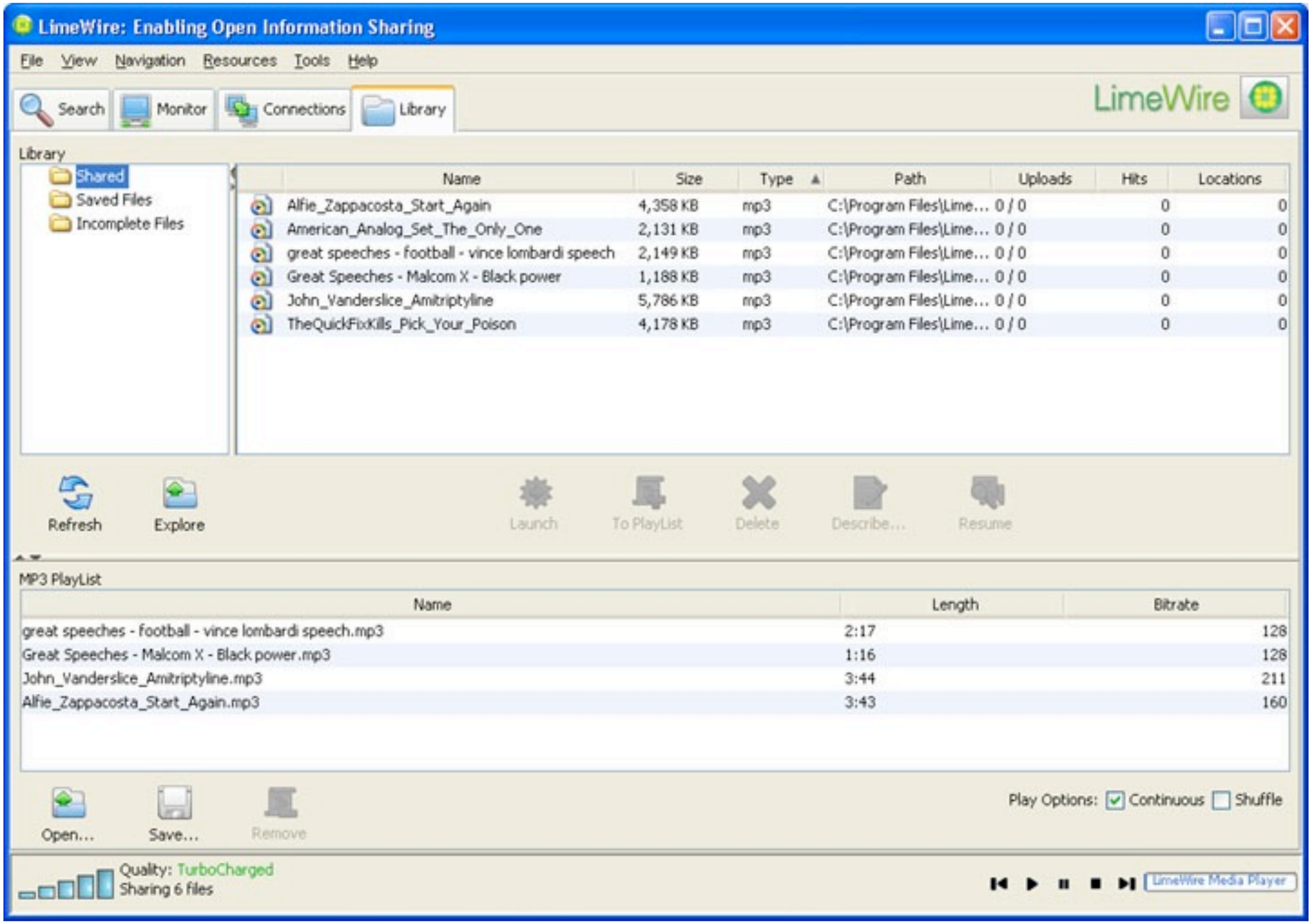


Gnutella

- **File Sharing**
  - Το (αρχικό) Napster κλείνει το 2001 μετά από απόφαση του δικαστηρίου για παράνομη ανταλλαγή αρχείων.
  - Στις αρχές του 2000 δημιουργείται από τους κατασκευαστές του Winamp, ένα νέο σύστημα στο οποίο δεν υπάρχει κεντρικοποιημένη διαχείριση της λίστας των αρχείων.
  - Με αυτό τον τρόπο δημιουργείται ένα εντελώς κατανεμημένο σύστημα (που θεωρητικά δεν μπορεί να κλείσει κανείς…)



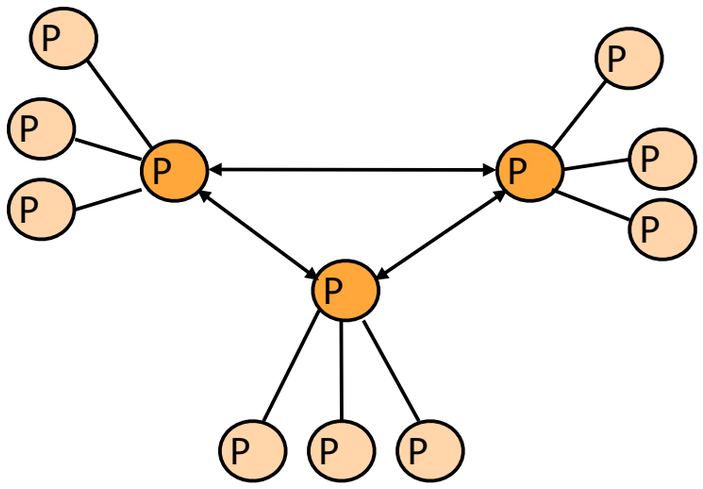
Παράθυρο Αναζήτησης Gnutella



Hybrid P2P

- **Skype : Διαδικτυακή Τηλεφωνία**

– Δωρεάν συνομιλία με άλλους χρήστες οπουδήποτε στον κόσμο.
– Δημιουργήθηκε από τους ιδρυτές του εργαλείου ανταλλαγής αρχείων KaZaA
– Η Αρχιτεκτονική Π/Π χρησιμοποιείται για να δρομολογεί έξυπνα τα πακέτα φωνής μεταξύ των διάφορων Πελατών

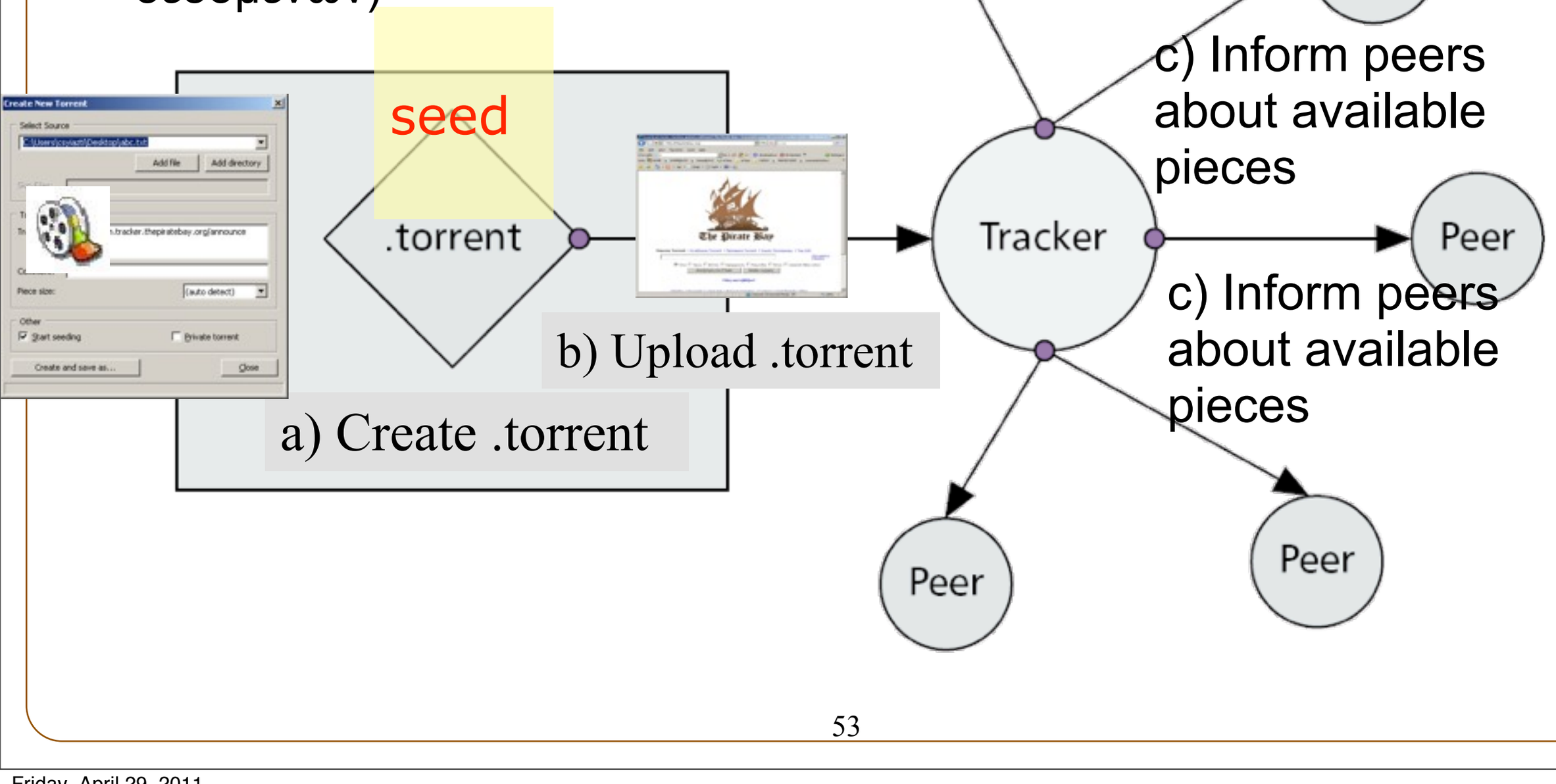


Το πρωτόκολλο επικοινωνίας των superpeers είναι κλειστό (proprietary)

Centralized P2P

BitTorrent: File Sharing

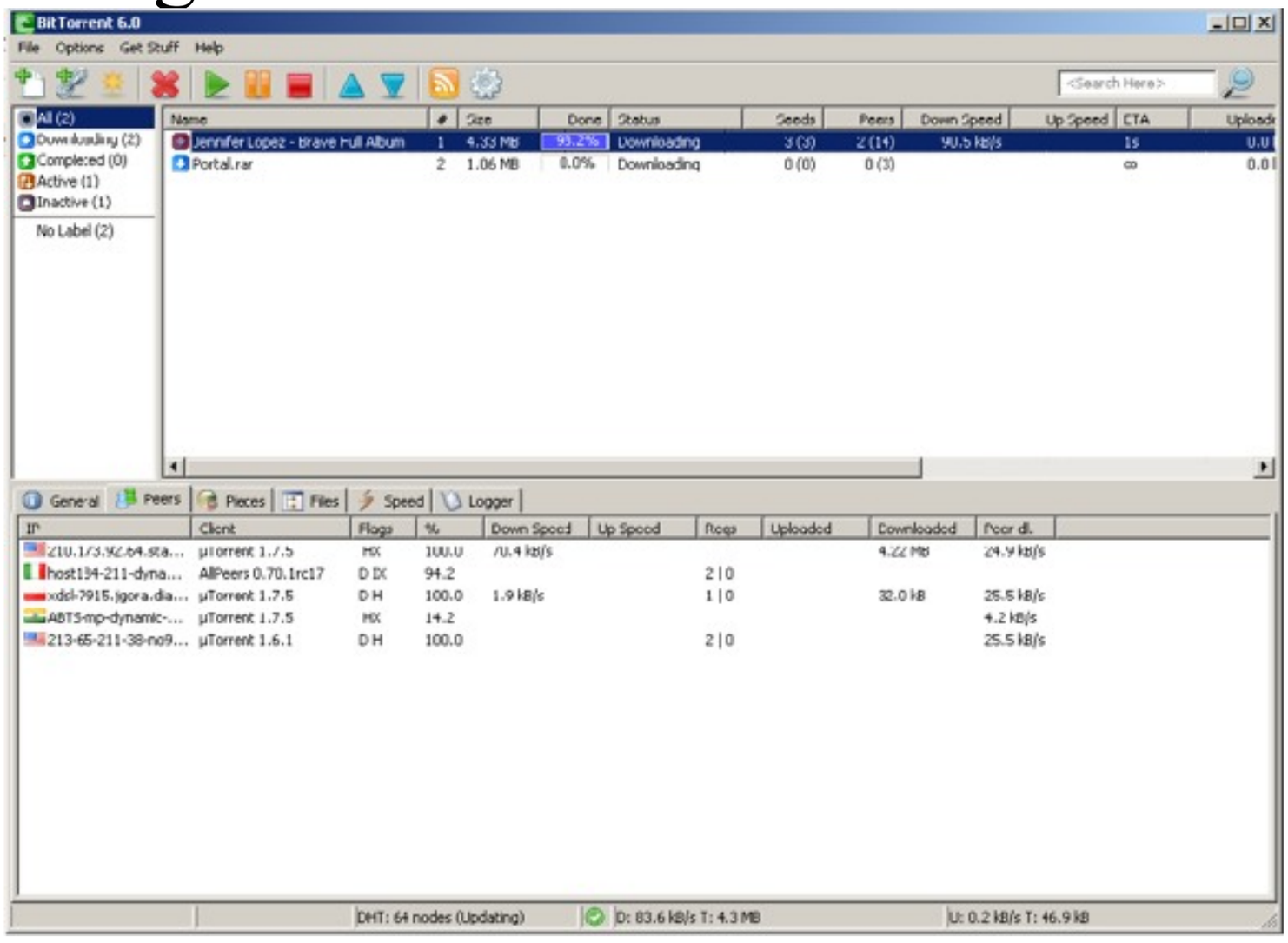
Ένα πρωτόκολλο μεταφοράς δεδομένων (όχι αναζήτησης δεδομένων)



Centralized P2P

BitTorrent: File Searching

Αναζήτηση



Ανάκτηση 256KB από κάθε Peer μέχρι να ανακτηθεί όλο το αρχείο. Τα κομμάτια που ανακτούνται γίνονται παράλληλα διαθέσιμα στους υπόλοιπους μέσω του tracker

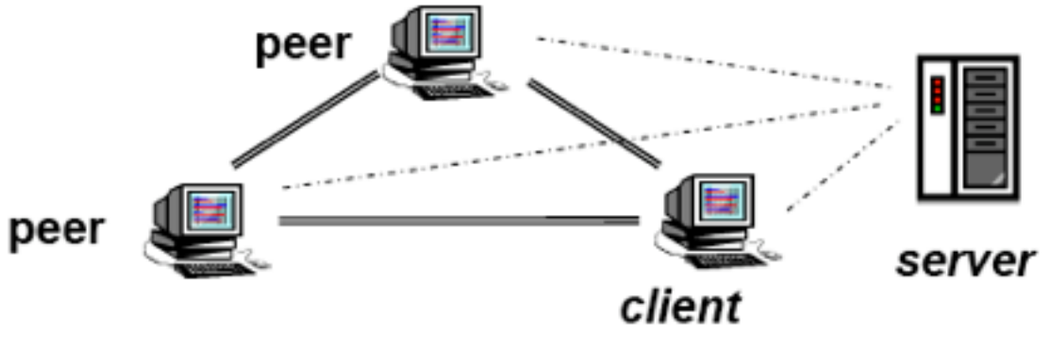
Mechanisms

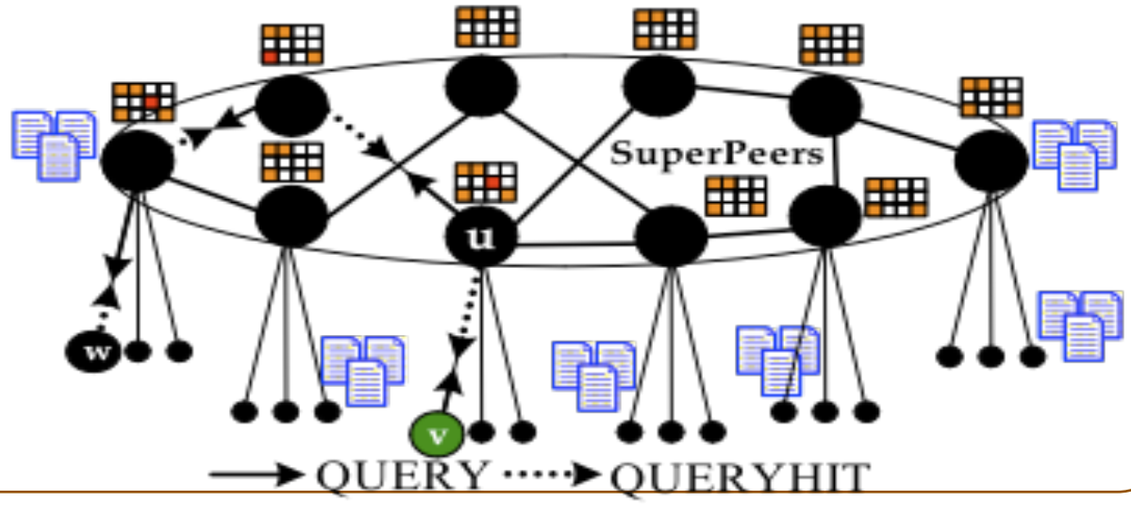
How do P2P systems work?

- Fundamental architectural choices:
  - Degree of centralization
  - Structure of the overlay network
- Key challenge:
  - Build an overlay with a routing capability that works well in the presence of a high membership turnover (**churn**)
- Key problems to meet the challenge:
  - Application state maintenance
  - Application-level node coordination
  - Content distribution

Degree of centralization

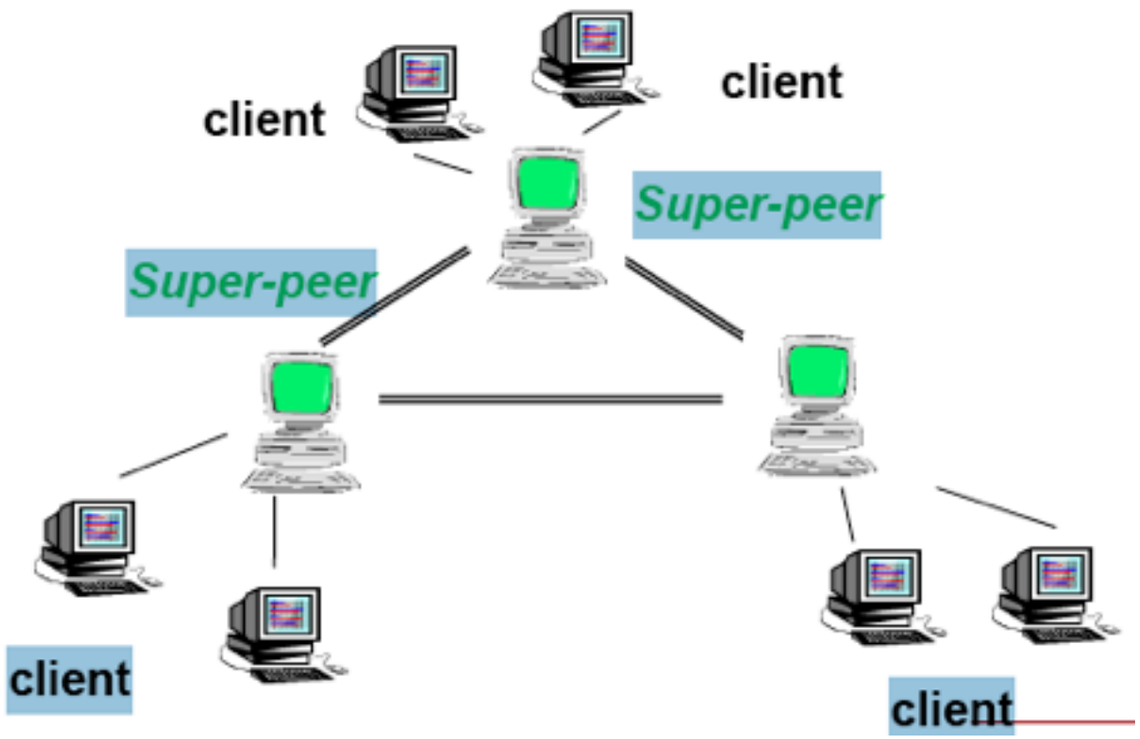
- Centralized
  - –Κεντρικά Ευρετήρια Αναζήτησης π.χ., Napster, Bittorent
- Purely Distributed
  - –Δεν υπάρχουν Ευρετήρια Αναζήτησης π.χ., Gnutella
- Hybrid
  - –Κάποιοι επιλεγμένοι peers (η επιλογή γίνεται βάση του διαθέσιμου bandwidth, της ώρα σύνδεσης, κτλ) έχουν μερικά ευρετήρια για τα περιεχόμενα άλλων κόμβων, π.χ., FastTrack (KaZaA), Limewire's Ultrapeers (Superpeers), Skype



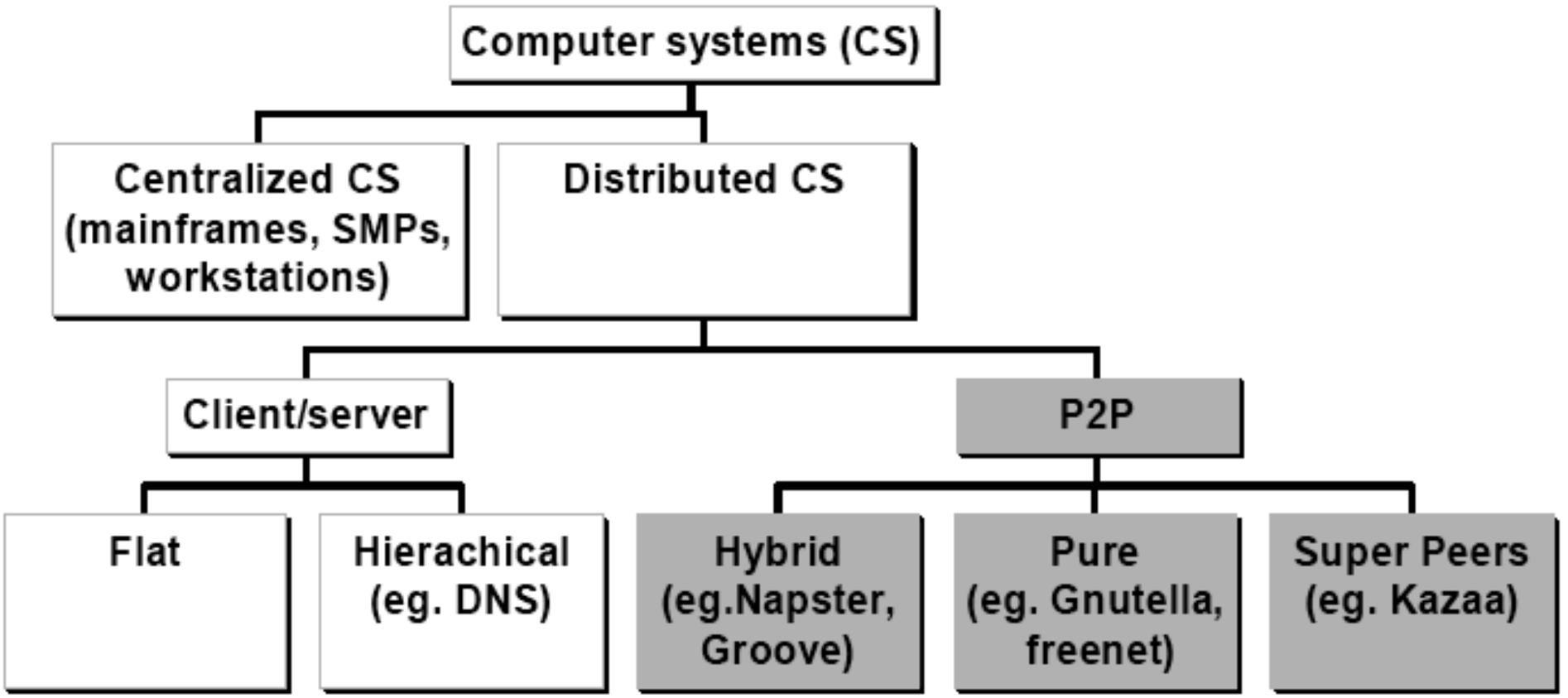


Super-peers

- A super-peer is a node in a peer-to-peer network that operates both as a server to a set of clients, and as an equal in a network of super-peers.
- Super-peer networks try to balance the efficiency of centralized search, and the autonomy, load balancing and robustness to attacks provided by distributed search.
  - example: Kazaa

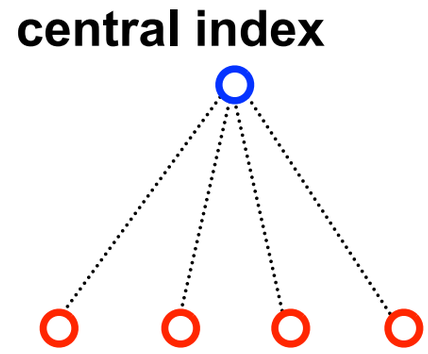


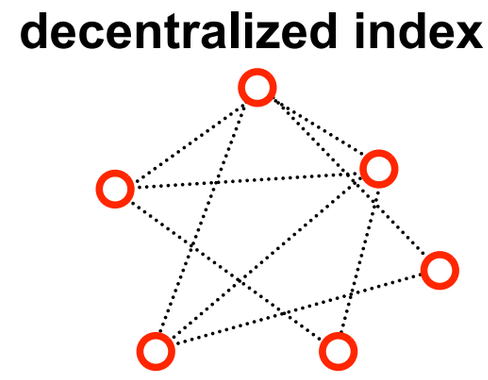
P2P classification



Motivation for structured P2P

- Peer-to-Peer file-sharing very popular
- Napster
  - Completely centralized
  - Central server knows who has what
  - Judicial problems
- Gnutella
  - Completely decentralized
  - Ask everyone you know to find data
  - Very inefficient
- Need decentralized solutions with more efficient index structures:
  - Structured overlays





The problem of Overlay Maintenance

- P2P systems maintain an overlay network, which can be though of as a graph G = (N,E) where:
  - N: set of participating computers
  - E: set of *overlay* links
  - Two nodes linked together, are aware of each other's IP address and communicate directly via the Internet

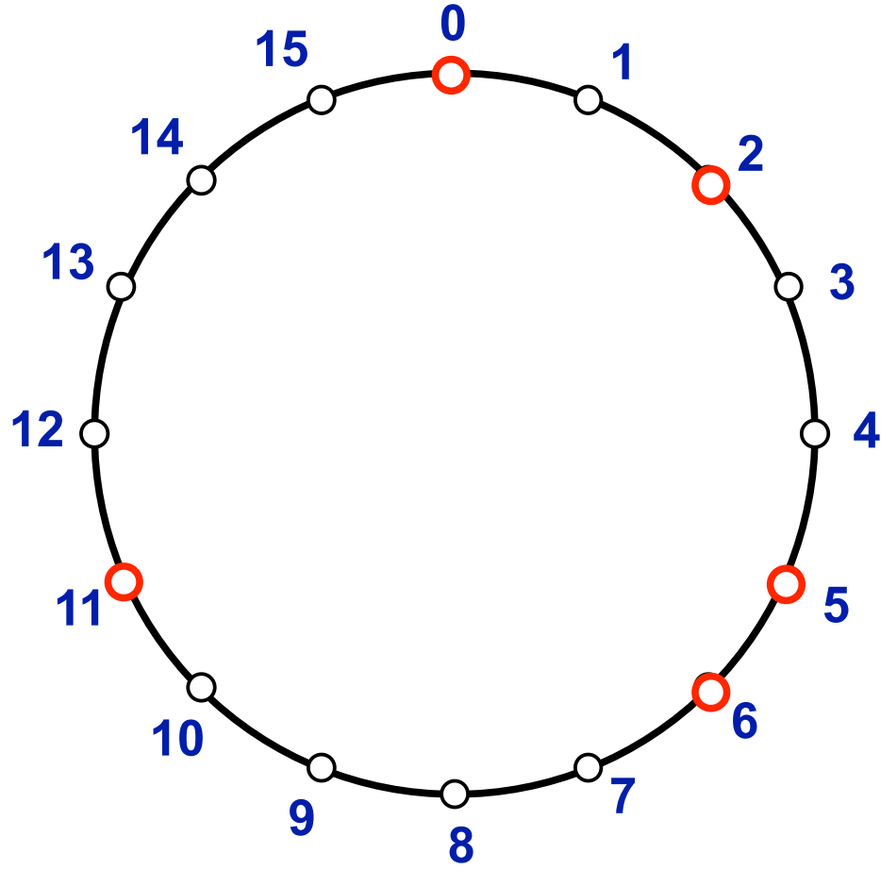
- Connection of new nodes:
  - In partly centralized systems:
    - connect first to controller node located at a well-known domain name or IP address.
    - initially a star-shaped topology
  - In decentralized overlays:
    - obtain, through an outside channel (Web), the network address of some *bootstrap* node already in the system

Structured vs Unstructured Overlays

- Unstructured Overlays:
  - No constraints on the links between different nodes
  - No particular graph structure
  - Minimum and maximum degree for each node (why?)
  - A node picks its neighbors by conducting a random walk on the graph, for each neighbor
- **Structured overlays**
  - Each node has a unique identifier in a large numeric key space (e.g. in the set of 160-bit integers)
  - Identifiers are chosen so that they are uniformly distributed in that space
  - A node's identifier determines its position within the structure of the graph and constrains its set of overlay links
  - Keys are used when assigning responsibilities to nodes

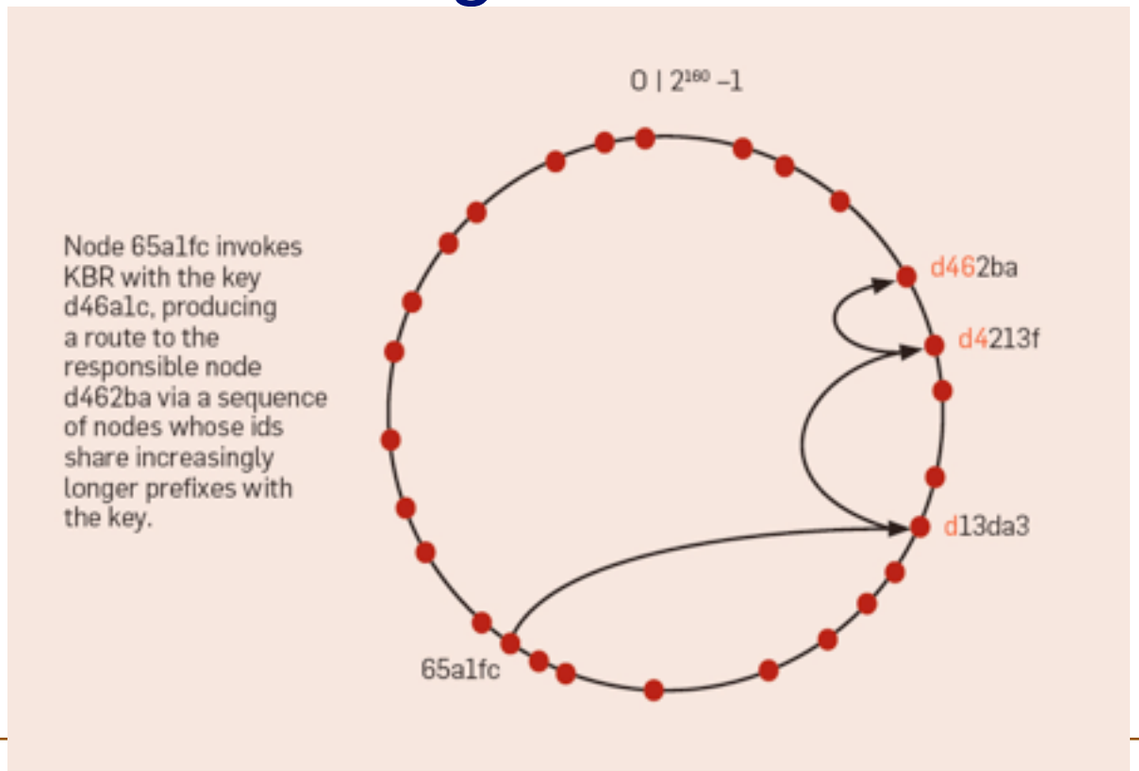
Structured Overlays

- Managing the key space:
  - the key space is divided among the participating nodes, such that each key is mapped to exactly one of the current overlay nodes via a simple function
  - E.g. a key may be mapped to the node whose identifier is the key's closest counterclockwise successor in the key space



Key-based routing

- Given a starting node n_0 and a key k, KBR produces a *path* that ends to the node responsible for k
  - –Path: a sequence of connected overlay nodes
  - –Different KBR implementations seek to strike a balance between the amount of routing state maintained at each node and the number of forwarding hops
  - –For typical implementations, the amount of routing state and the number of forwarding hops are both *logarithmic* with respect to network size

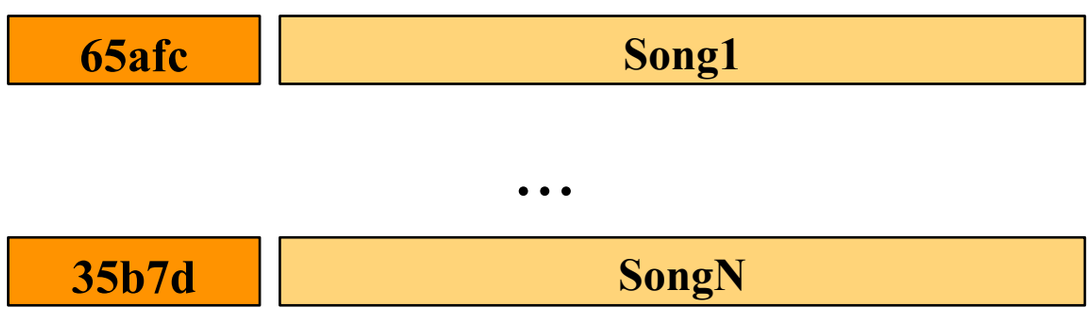


KBR and Structured P2P Overlays

- Structured P2P:
  - –invest additional resources at each node in order to maintain a specific graph structure which can perform KBR
  - –KBR can reliably and efficiently locate uniquely identified data items and maintain spanning trees among member nodes
- Maintaining a structured overlay in a high-churn environment has an associated cost, which may not be worth paying if the application does not require the functionality provided by key-based routing.
- Choice between structured and unstructured depends on:
  - –how useful KBR is for the driving application
  - –the amount of churn

Distributed state

- Most P2P systems maintain some application-specific distributed state:
  - State: collection of objects with unique keys.
- A key goal for P2P systems is to maintain this collection of state objects in a distributed manner by providing distributed mechanisms for:
  - object placement
  - object discovery



Distributed state in Partly centralized P2P

- Placement:
  - –An object is typically stored at the P2P node that inserted the object, as well as at any nodes that have subsequently downloaded the object.
- Discovery:
  - –The controller node maintains information about which objects exist in the system, their keys, names and other attributes, and which nodes are currently storing those objects.
  - –Queries for a given key, or a set of keywords that match an object's name or attributes, are directed to the controller, which responds with a set of nodes from which the corresponding object(s) can be downloaded.

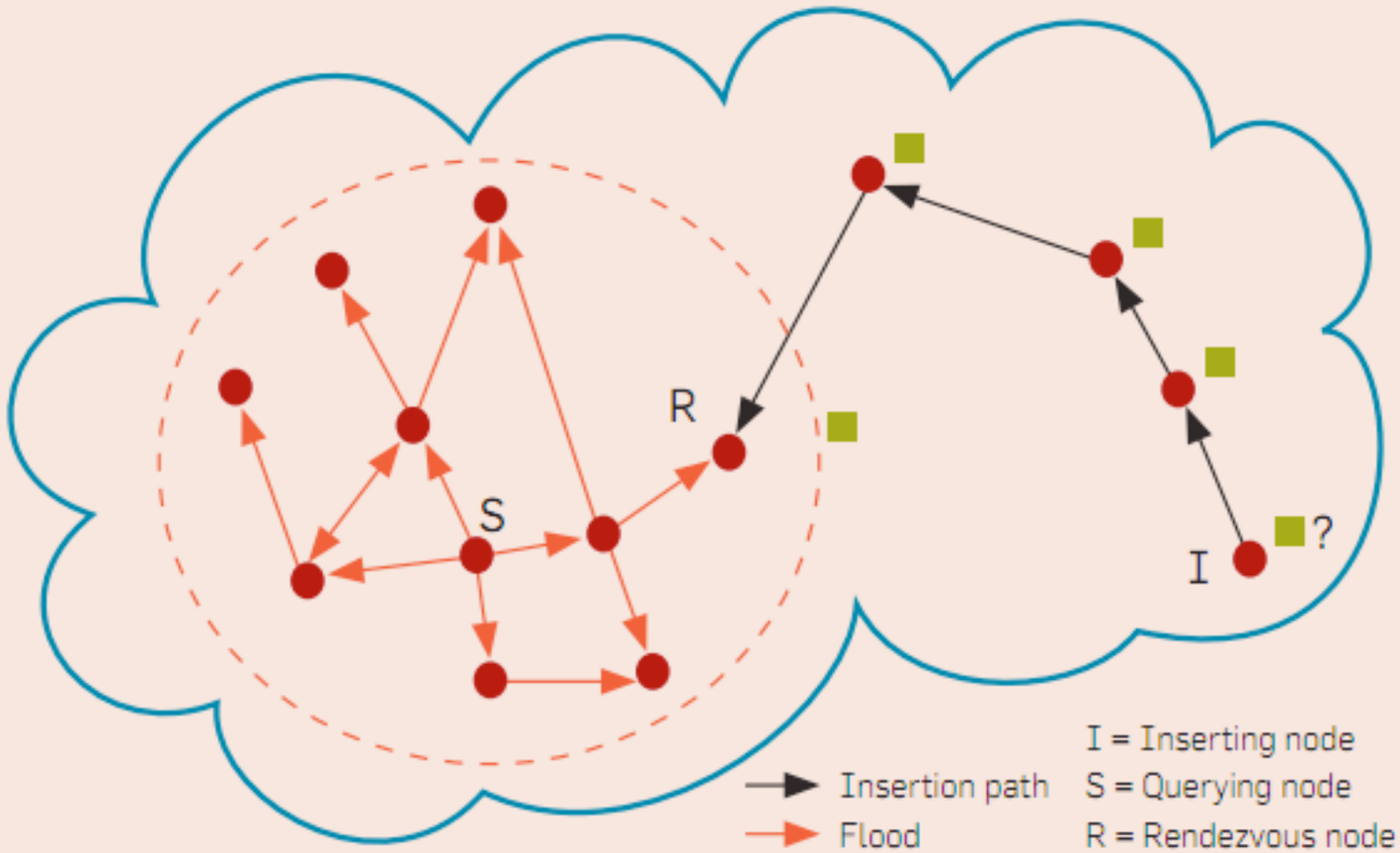
Distributed state in Unstructured P2P

- Placement:
  - –Content is typically stored at the node that introduced the content to the system, and replicated at other downloaders.
  - –Some systems place copies of (or pointers to) an inserted object on additional nodes, for instance, along a random walk path through the overlay.

- Discovery:
  - –To locate an object, a querying node typically floods a request message through the overlay.
    - The query can specify the desired object by its key, metadata, or keywords.
  - –A node that receives a query and has a matching object (or a pointer to a matching object), responds to the querying node.

Flooding

Figure 2. Locating objects in unstructured overlays.



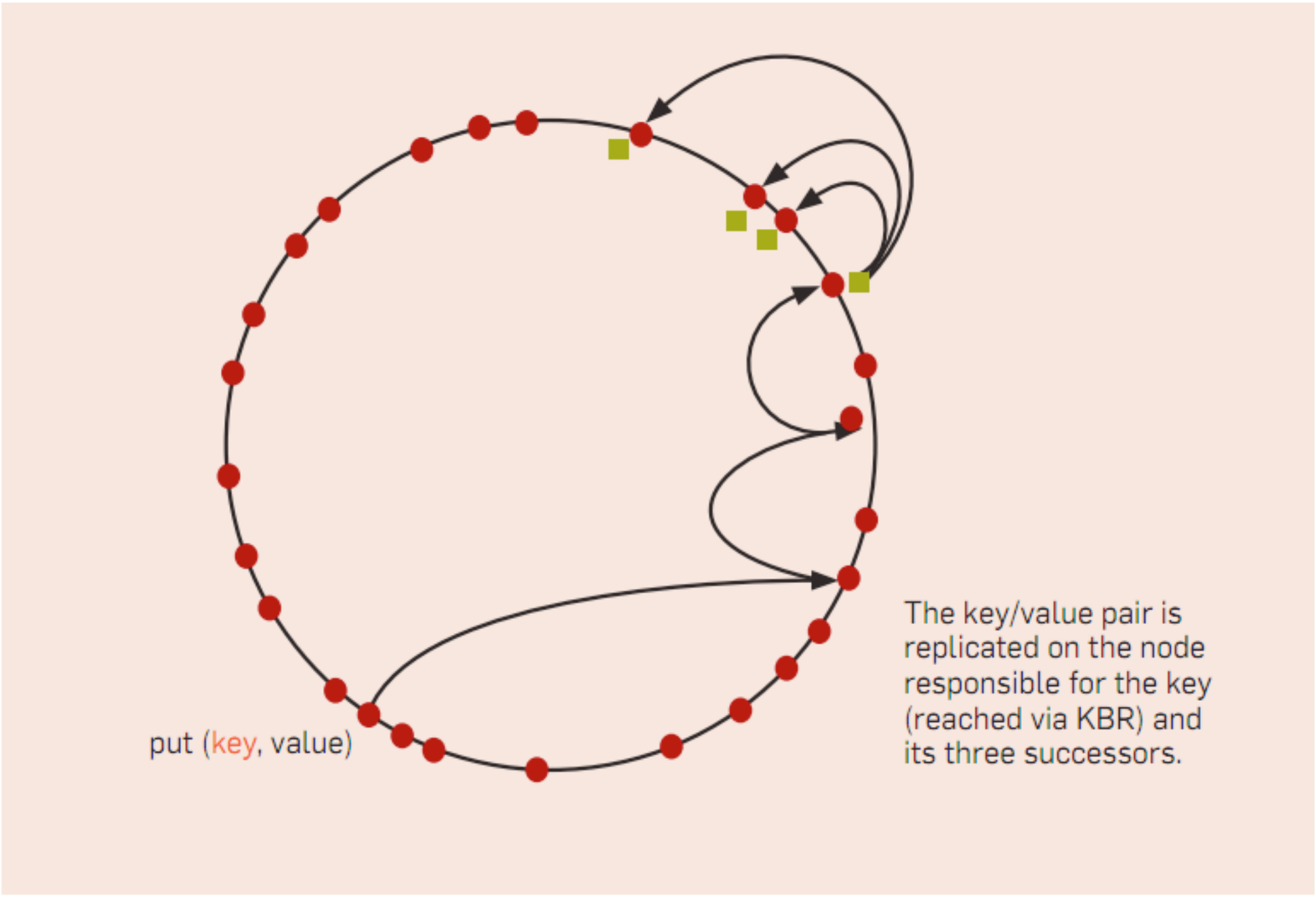
Node I adds and advertises the green object by inserting pointers to the green object on all nodes along a random walk through the overlay. When node S tries to locate the green object, it floods a query through the overlay. When the query reaches node R, R returns the address of I.

- Often, the scope of the flood (that is,the maximal number of hops from the querying nodes that a flood message is forwarded) is limited to trade recall (the probability that an object that exists in the system is found) for overhead (the number of messages required by the flood).
- An alternative to flooding is for the querying node to send a request message along a random walk through the overlay

Distributed state in Structured P2P

- State maintenance is performed using the Distributed Hash Table (DHT) abstraction.
- The DHT has the same put/get interface as conventional hash tables, but:
  - inserted (key,value) pairs are distributed among the participating nodes in the structured overlay using a simple placement function:
    - "values" are the state objects maintained by the system
- Example of a placement function: **place replicas of the key/value pair on the set of r nodes whose identifiers succeed the key in the circular key space**
  - This function can be implemented using the KBR primitive
  - To insert (put) a value: use the KBR primitive to determine the node responsible for key k, and store the pair on that node, which then propagates it to the set of replicas for k.
  - To lookup (get) a value: use KBR to fetch the value associated with a given key

put in DHT


put (key, value)
The key/value pair is replicated on the node responsible for the key (reached via KBR) and its three successors.

Churn

- What does it mean?
- When a DHT experiences churn, pairs have to be moved between nodes as the mapping of keys to nodes changes
- To avoid large data transfers during churn, large data values are typically *not* inserted directly into a DHT; instead an indirection pointer is inserted under the value's key, which points to the node that actually stores the value

Structured vs Unstructured

- Unstructured overlays:
  - –tend to be very efficient at locating widely replicated objects - finding “hay”
  - –support arbitrary keyword-based queries
- KBR-based techniques:
  - –can reliably and efficiently locate any object that exists in the system, no matter how rare it may be - finding “needles”
  - –directly support only key-based queries

Distributed Coordination in P2P

- A group of nodes in a P2P application must coordinate their actions without centralized control. E.g.:
  - A set of nodes that replicate a particular object must inform each other of updates to the objects
  - A node interested in receiving a particular streaming content channel may wish to find, among the nodes that currently receive the channel, one that is nearby and has available upstream network bandwidth
- Coordination approaches:
  - Epidemic: information spreads *virally*
  - Tree-based: distribution trees are formed to spread the info
  - Centralized: coordination is accomplished through the controller

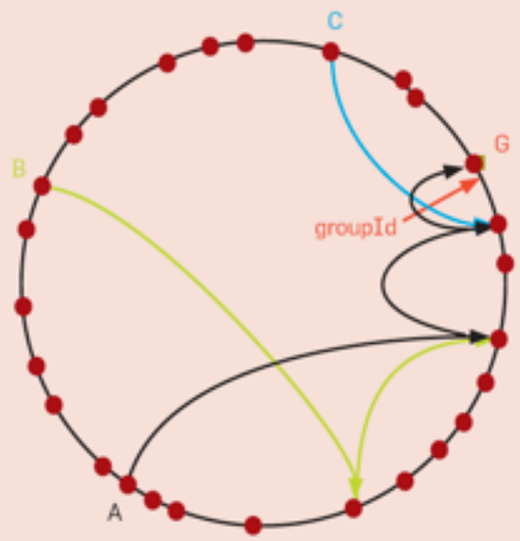
Distributed Coordination in unstructured P2P

- Epidemic dissemination
  - –Best suited for unstructured overlays
  - –Simple and robust
  - –There is a trade-off between the speed of information dissemination and overhead
    - The information ends up being needlessly delivered to all nodes
- Spanning-tree (γεννητορικό δένδρο) dissemination
  - –ST connects the nodes that should be coordinated
  - –The ST is embedded in the overlay graph using a *decentralized algorithm* for ST construction
  - –The ST is used to multicast messages to all members or to compute summaries (sums, averages, minima, maxima) of state variables within the group
  - –Added efficiency should be balanced against overhead of maintaining a spanning tree in the unstructured overlay network

Distributed Coordination in structured P2P

- Spanning trees can be formed and maintained very efficiently using the KBR primitive
  - ➡trees is the preferred method of coordination for structured overlays
- To join a ST, a node uses KBR to route to a unique key associated with a coordination group
  - –The resulting union of the KBR paths from all group members is a ST rooted at the node responsible of the group's key
- The KBR tree is used to aggrega
  state associated with the group,
  multicast and anycast.

Figure 4. An example KBR tree.



The KBR routes from group member nodes A, B, and C to G (the node responsible for the group key) form a spanning tree rooted at G.

Scalability

- Why is the Spanning Tree approach scalable?
  - –A join message *terminates* as soon as it intercepts the Spanning Tree
    - Tree maintenance is decentralized: arrival and departure of a node is noted only by the node's parent and children in the tree
  - ➡The technique scales to:
    - large numbers of groups
    - large and highly dynamic groups

Content Distribution

- A common task for P2P systems: distribute bulk data or streaming content
- Techniques:
  - –Based on fixed distribution trees: either formed on structured overlays or embedded in unstructured overlays
  - –Swarming protocols: no notion of fixed tree for routing content - followed in Bittorent
    - Content divided into blocks
    - Each block is individual multicast to all overlay nodes, such that different blocks are disseminated along different paths
    - Once every swarming interval, overlay neighbors exchange information indicating which content blocks they have available
    - Each node requests from its neighbors a block that it does not already have

Challenges

- Controlling membership
  - –sybil attacks: attacker creates many distinct identities to populate a P2P system with nodes under his control
    - Proof of work required by a node before joining the overlay
    - Certified identities
- Protecting data: availability, durability, integrity, and authenticity of data
  - –Self-certifying objects; peers vote on the authenticity of data
  - –Replication to address availability under churn
    - A practical system cannot simultaneously achieve all three goals of scalable storage, high availability and resilience to churn.
- Incentives
  - –Tit-for-tat strategies to deal with free riding

Challenges (ctd)

- Managing P2P systems
  - –P2P systems adapt to a wide range of conditions wrt workload and resource availability; recover from most failures; participating users look after their hardware independently
  - –However, the lack of centralized control makes it difficult to manage systemwide disruptions when they occur