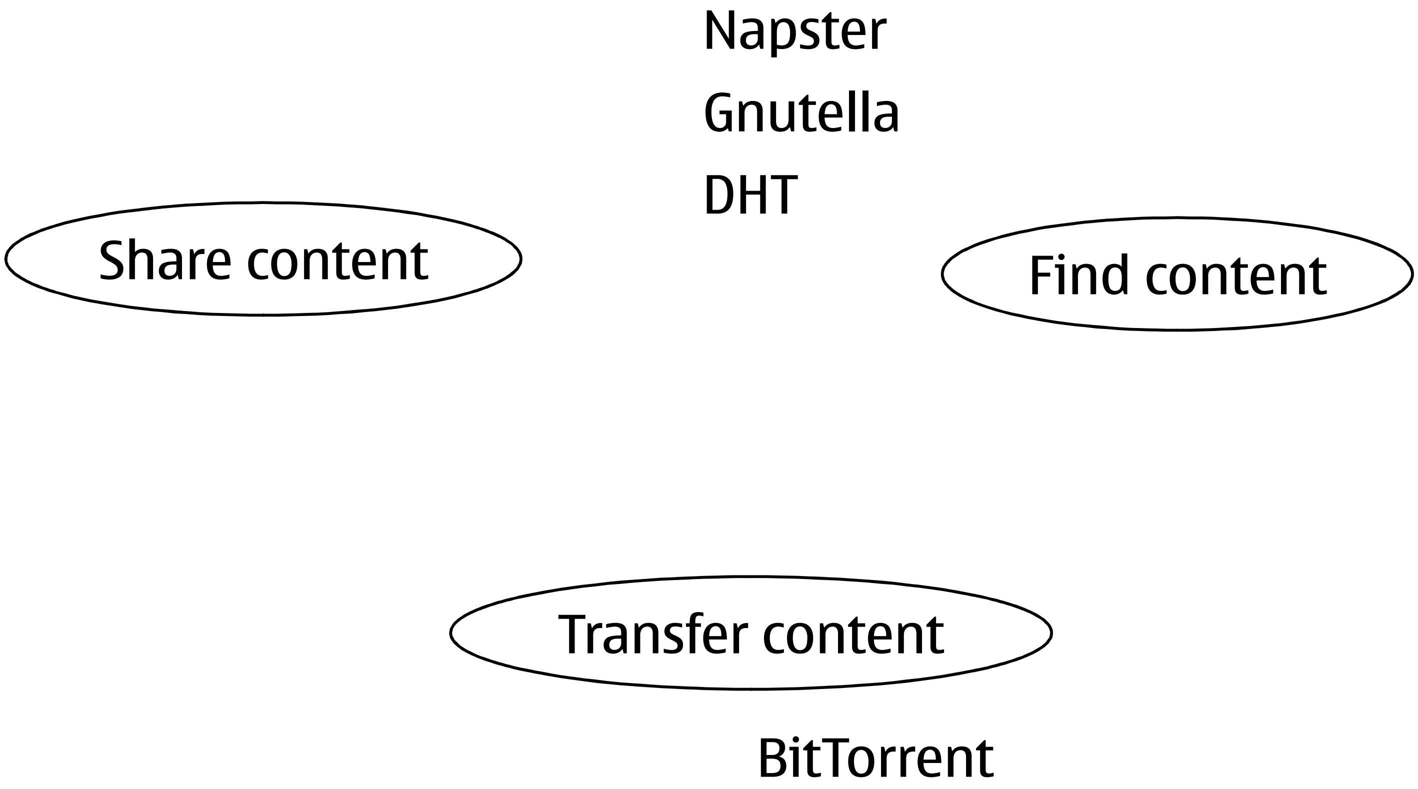
Content Search

Unstructured P2P

Jukka K. Nurminen

*Partly adapted from original slides provided by Rüdiger Schollmeier and Jörg Eberspächer (Technische Universität München)

Steps of content sharing



Napster

Gnutella

DHT

Share content

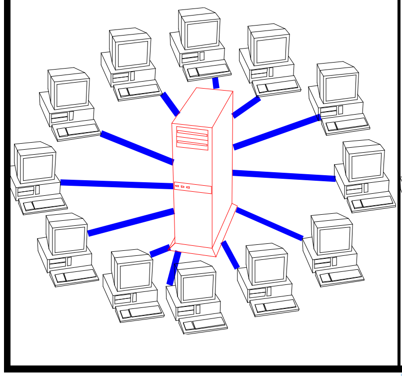
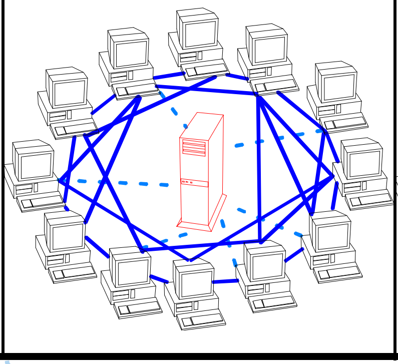
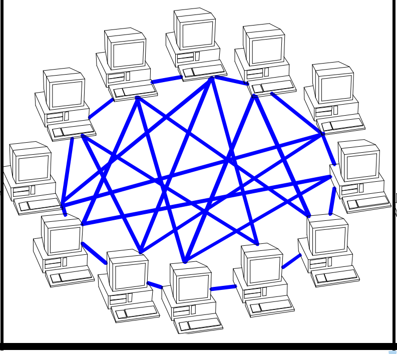
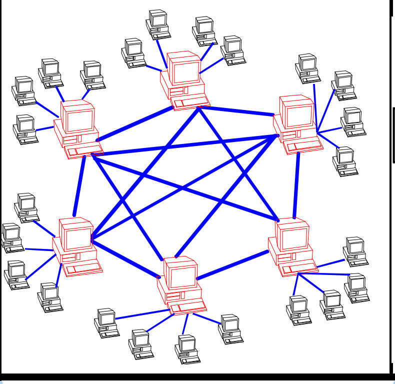
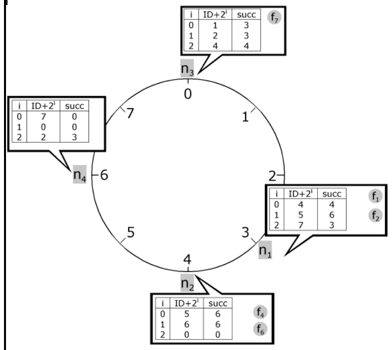
Find content

Transfer content

BitTorrent

X.1 The Architectures of 1st and 2nd Gen. P2P

<table>
<tr><th><em>Client-Server</em></th><th colspan="4"><em>Peer-to-Peer</em></th></tr>
<tr><td rowspan="4">1. Server is the central entity and only provider of service and content. → Network managed by the Server
2. Server as the higher performance system.
3. Clients as the lower performance system

Example: WWW</td><td colspan="4">1. Resources are shared between the peers
2. Resources can be accessed directly from other peers
3. Peer is provider and requestor (Servent concept)</td></tr>
<tr><td colspan="3"><strong><em>Unstructured P2P</em></strong></td><td><strong><em>Structured P2P</em></strong></td></tr>
<tr><td><em>Centralized P2P</em></td><td><em>Pure P2P</em></td><td><em>Hybrid P2P</em></td><td><em>DHT-Based</em></td></tr>
<tr><td>1. All features of Peer-to-Peer included
2. Central entity is necessary to provide the service
3. Central entity is some kind of index/group database
Example: Napster</td><td>1. All features of Peer-to-Peer included
2. Any terminal entity can be removed without loss of functionality
3. → No central entities
Examples: Gnutella 0.4, Freenet</td><td>1. All features of Peer-to-Peer included
2. Any terminal entity can be removed without loss of functionality
3. → dynamic central entities
Example: Gnutella 0.6, JXTA</td><td>1. All features of Peer-to-Peer included
2. Any terminal entity can be removed without loss of functionality
3. → No central entities
4. Connections in the overlay are “fixed”
Examples: Chord, CAN</td></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table><tr><th>i</th><th>$ID+2^i$</th><th>succ</th></tr><tr><td>0</td><td>1</td><td>3</td></tr><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>2</td><td>4</td><td>4</td></tr></table>
<table><tr><th>i</th><th>$ID+2^i$</th><th>succ</th></tr><tr><td>0</td><td>7</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td></tr></table>
<table><tr><th>i</th><th>$ID+2^i$</th><th>succ</th></tr><tr><td>0</td><td>4</td><td>4</td></tr><tr><td>1</td><td>5</td><td>6</td></tr><tr><td>2</td><td>7</td><td>3</td></tr></table>
<table><tr><th>i</th><th>$ID+2^i$</th><th>succ</th></tr><tr><td>0</td><td>5</td><td>6</td></tr><tr><td>1</td><td>6</td><td>6</td></tr><tr><td>2</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

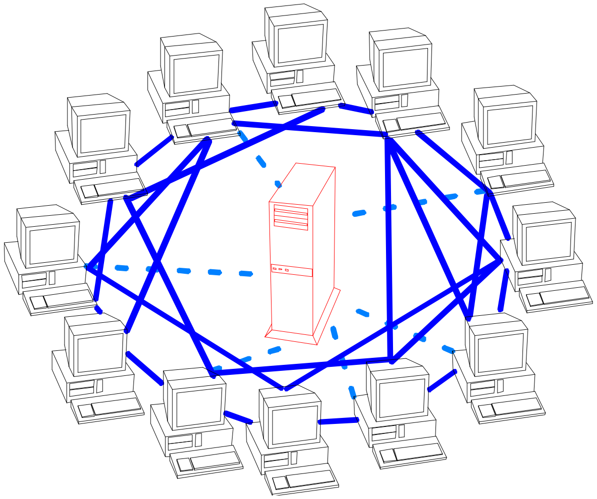
1st Gen. **2nd Gen.**

Further reading

- Ralf Steinmetz, Klaus Wehrle (Eds.): Peer-to-Peer Systems and Applications. Lecture Notes in Computer Science, Volume 3485, Springer, Berlin 2005
- Available (also electrically) at TKK library

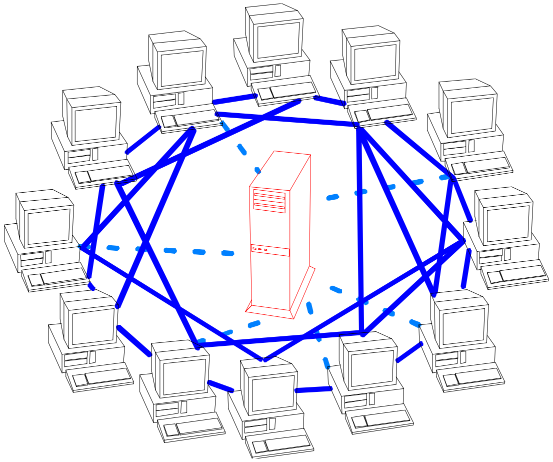
X. Overview

1. Centralized Peer-to-Peer Networks
2. Pure Peer-to-Peer Networks
3. Hybrid Peer-to-Peer Networks



X.2 Definition of centralized P2P

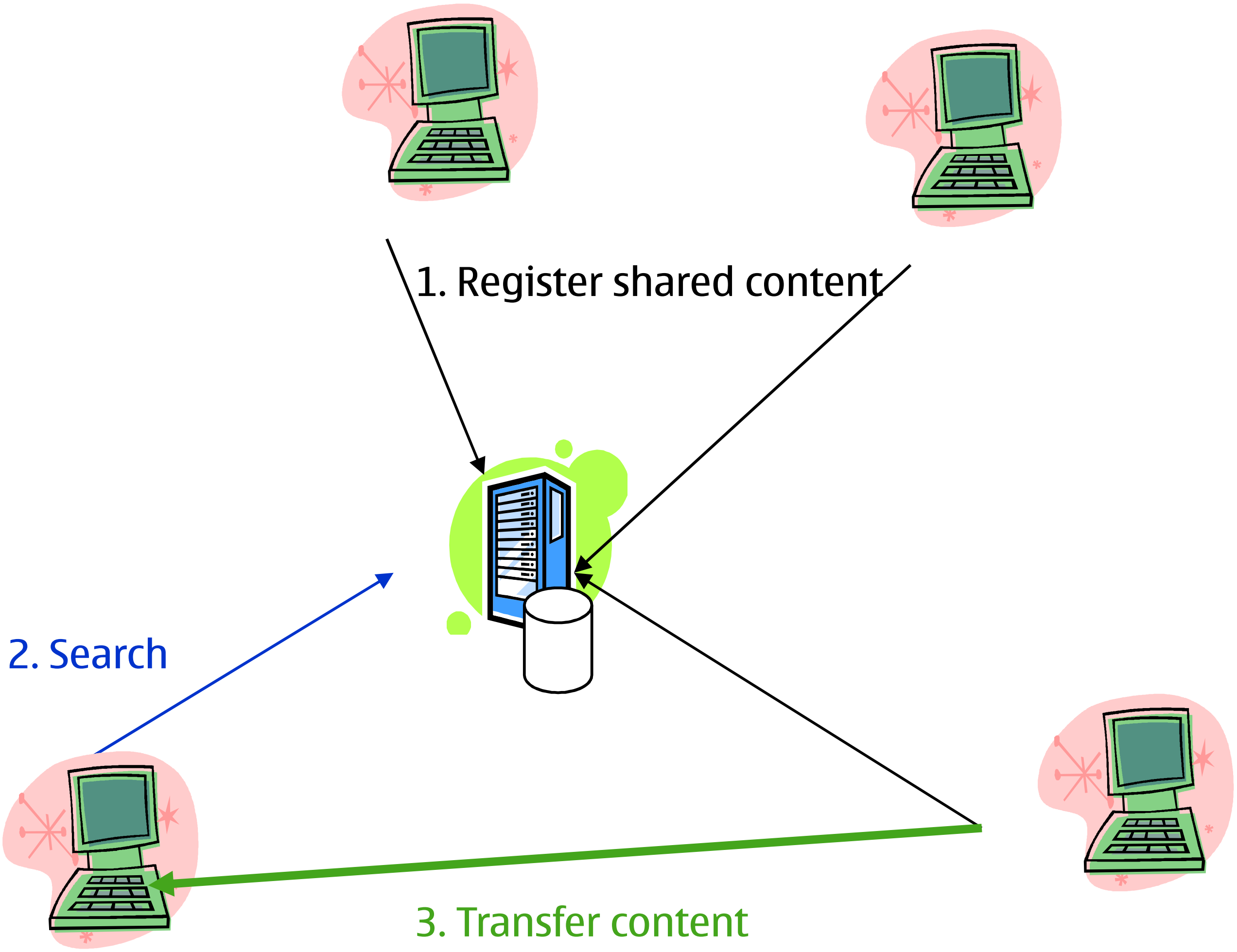
- All peers are connected to central entity
- Peers establish connections between each other on demand to exchange user data (e.g. mp3 compressed data)
- Central entity is necessary to provide the service
- Central entity is some kind of index/group database
- Central entity is lookup/routing table



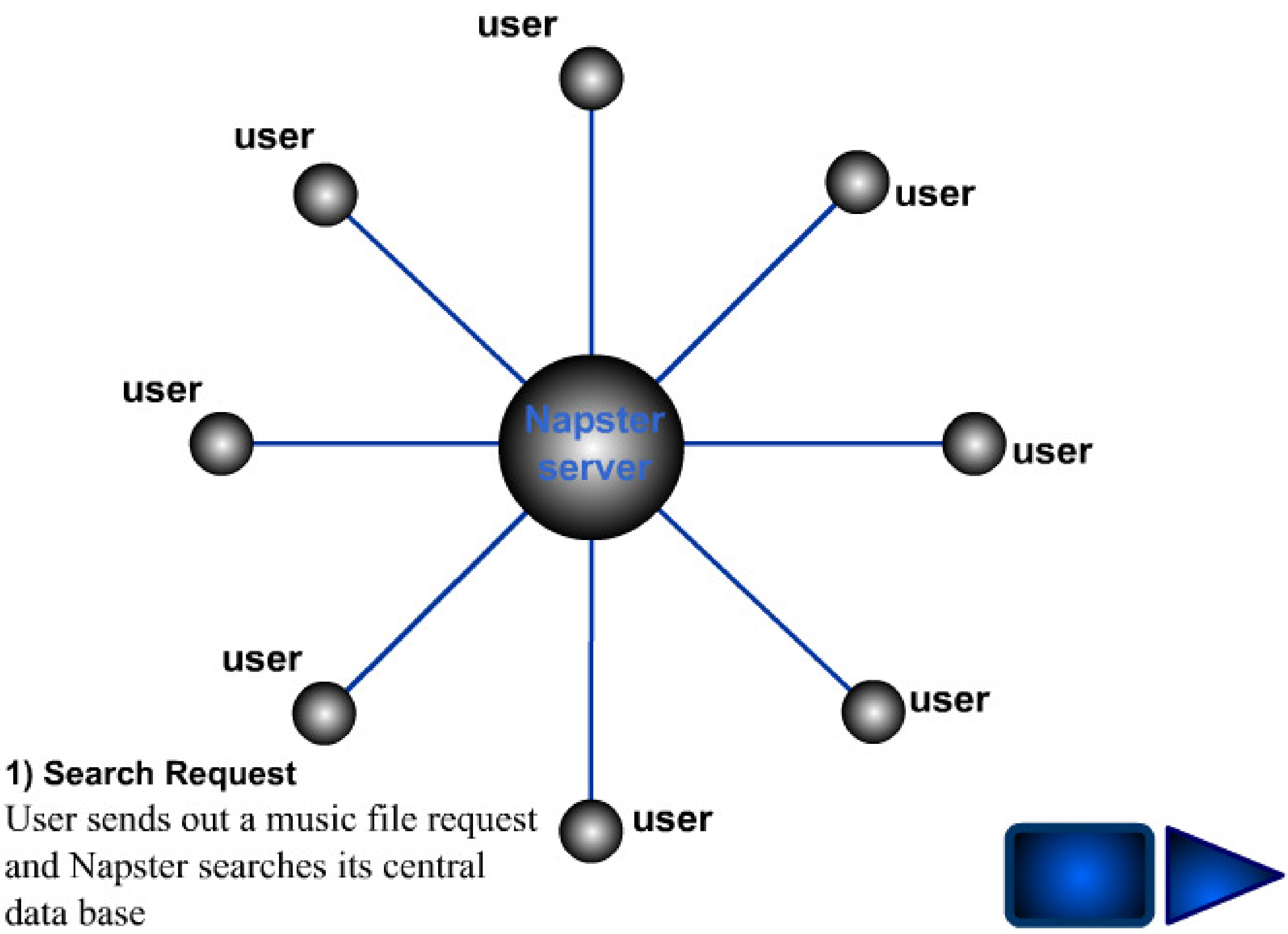
X.2 Example: Napster

- Program for sharing files over the Internet
- A “disruptive” application/technology?
- Brief History:
  - **May 1999**: Shawn Fanning (freshman, Northeastern University) founds Napster Online music service
  - **December 1999**: First Lawsuit
  - **March 2000**: University of Wisconsin reports that 25% of its IP traffic is Napster traffic
  - **December 2000**: estimated 60 million users
  - **February 2001**: US Circuit Court of appeals: napster knew users violating copyright laws
    →Shut down of the service

Napster



X.2 Napster animation



1) Search Request

User sends out a music file request and Napster searches its central data base

X.2 Discussion

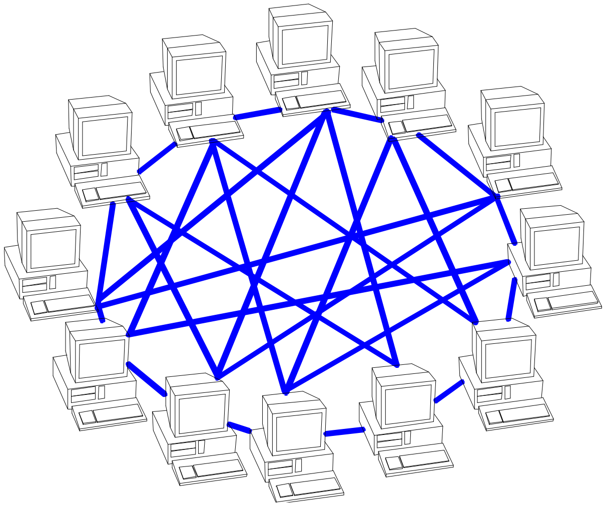
- Advantages
  - Simple
  - Efficient, little load for peers => also weak peers (mobiles) able to participate
  - Fast and complete lookup (one hop lookup)
  - Central managing/trust authority
  - Advertising business model
  - Clear legal responsibility

- Disadvantages
  - Single Point of Failure →easily attackable
  - Bottleneck
  - Potential of congestion

- Other applications and application areas
  - BiTorrent
  - VoIP (SIP, H.323)

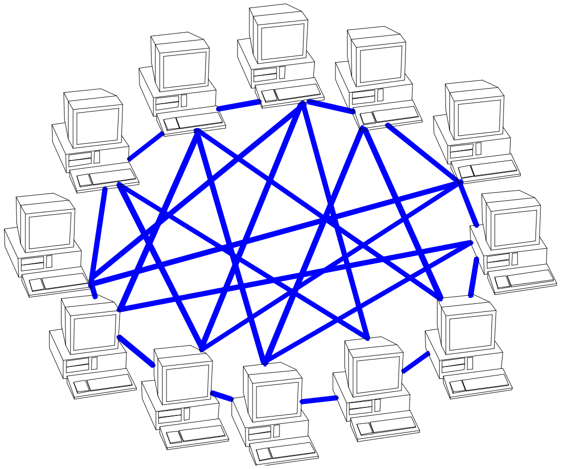
X. Overview

1. Centralized Peer-to-Peer Networks
2. Pure Peer-to-Peer Networks
3. Hybrid Peer-to-Peer Networks



X.3 Definition of Pure P2P

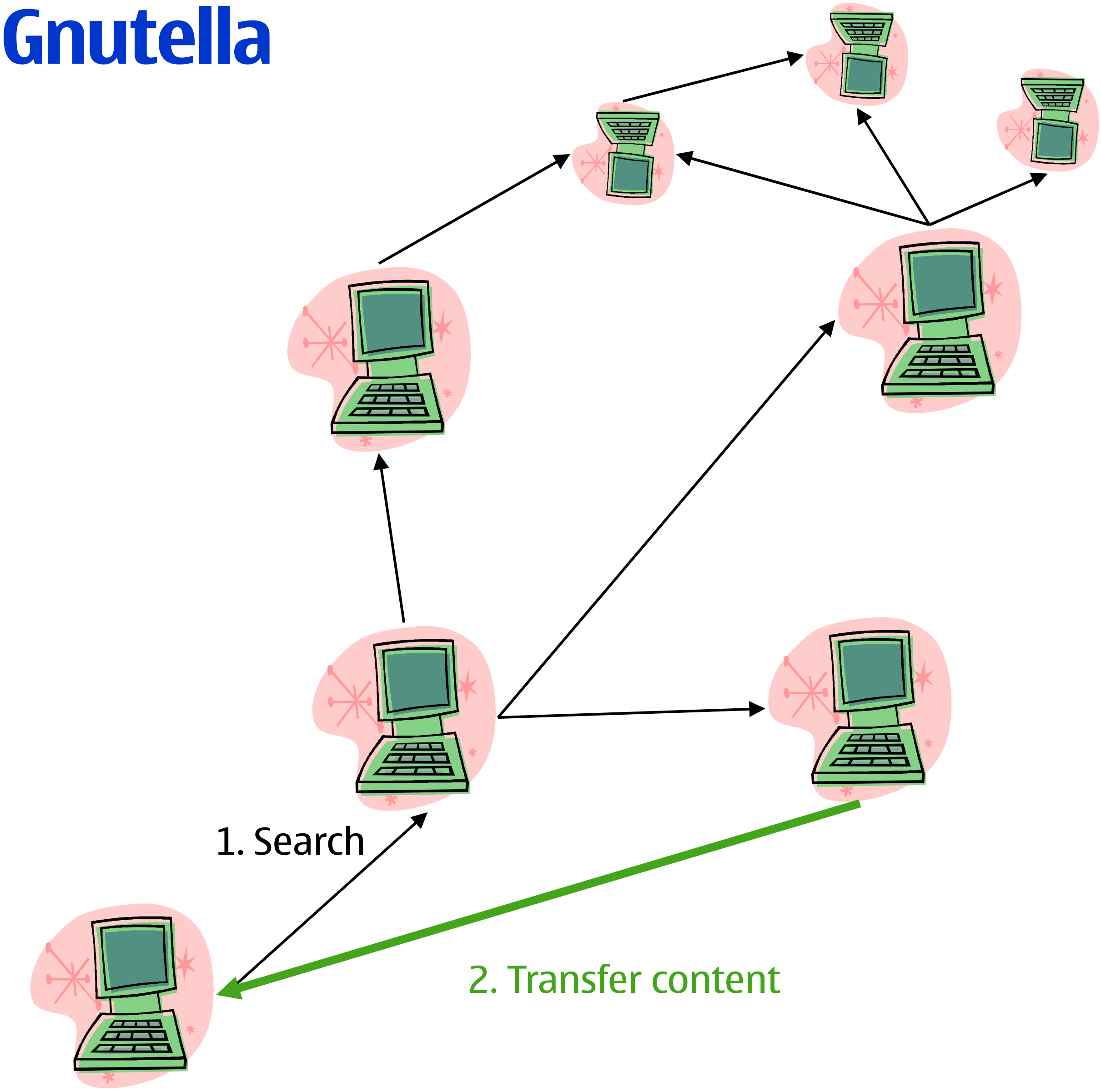
- Any terminal entity can be removed without loss of functionality
- No central entities employed in the overlay
- Peers establish connections between each other randomly
  - To route request and response messages
  - To insert request messages into the overlay



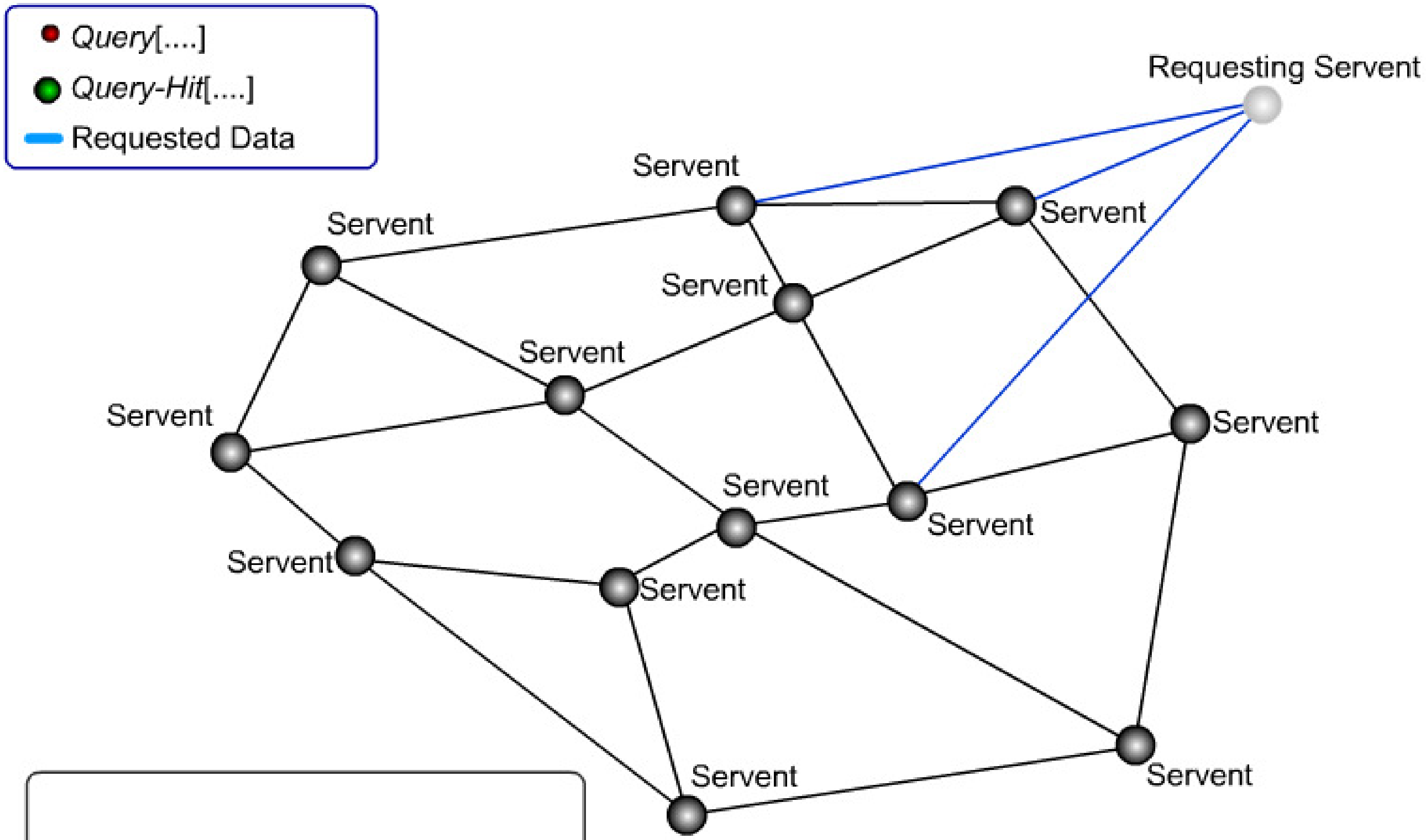
X.3 Example: Gnutella 0.4

- Program for sharing files over the Internet
- Focus: decentralized method of searching for files
- A “disruptive” application/technology?
- Brief History:
    - **March 2000**: open source release by by Justin Frankel and Tom Pepper of Nullsoft, a division of AOL, and almost immediately withdrawn
    - **Spring 2001**: further developments to improve scalability → Gnutella 0.6 (Hybrid P2P)
    - Since then:
        - available in a lot of implementations (Limewire, bearshare,...)
        - Developed further on (privacy, scalability, performance,...)

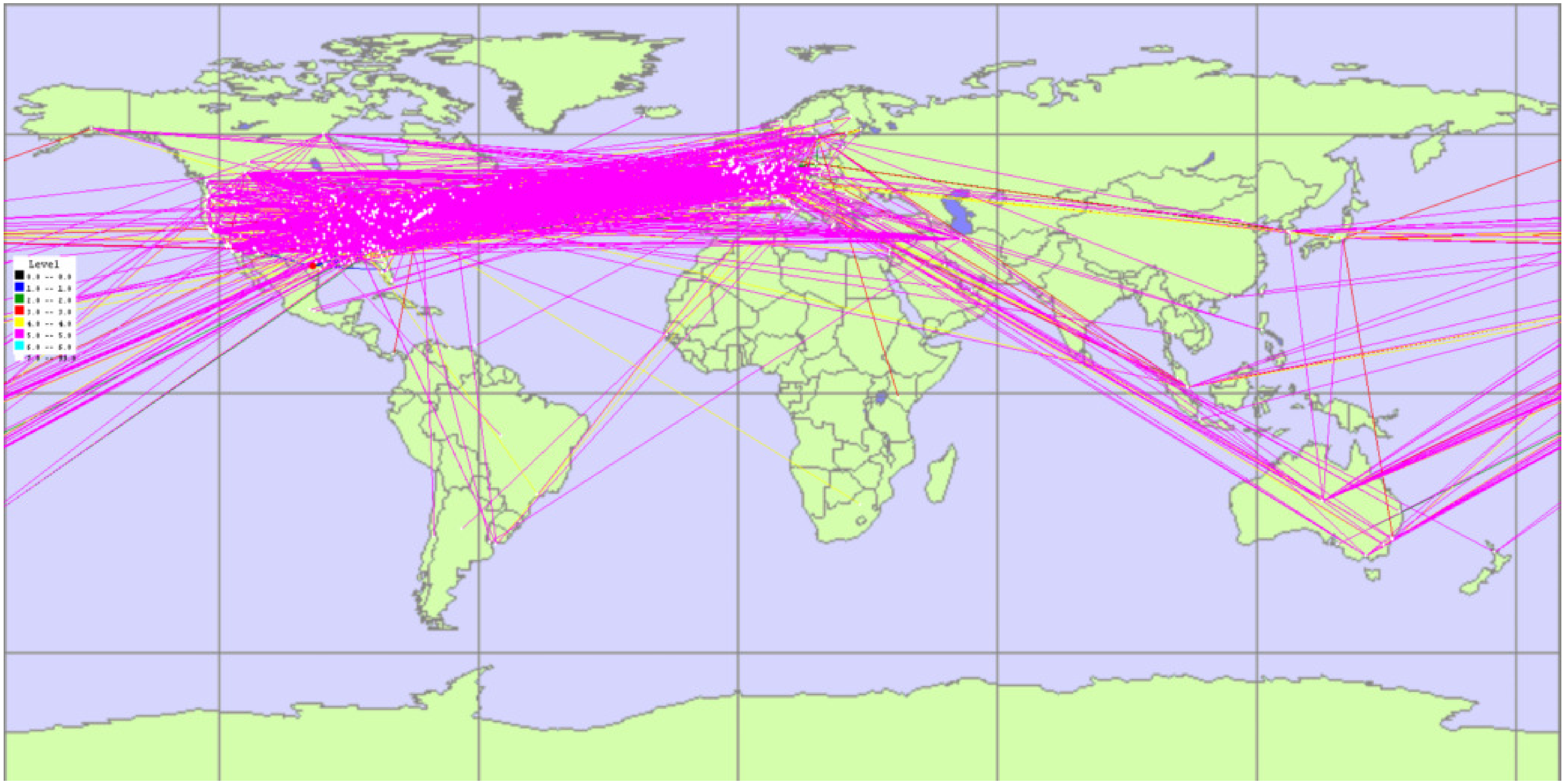
Gnutella



Gnutella Animation



X.3 The Gnutella Network



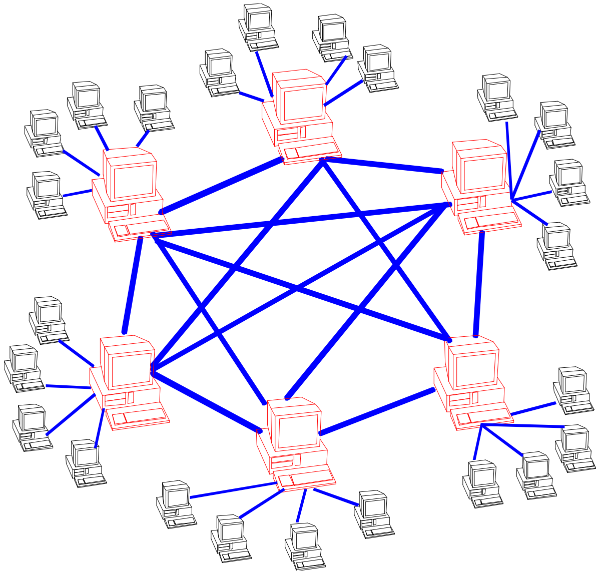
Measurements taken at
the LKN in May 2002

X.3 Discussion

- Advantages
    - Genuine P2P
    - Fully distributed, no servers
    - No single point of failure
    - Can provide anonymity
- Disadvantages
    - Flooding creates a lot of traffic
    - Unequal distribution of load
    - No responsible party
        - Business model?
        - Legal responsibility?
    - Overlay topology not optimal, as
        - no complete view available,
        - no coordinator
        - Zigzag routes, loops
        - If not adapted to physical structure delay and total network load increases
- Other applications and application areas
    - Freenet
        - Focus on anonymity and privacy
        - Content is transferred in chunks over the signaling network

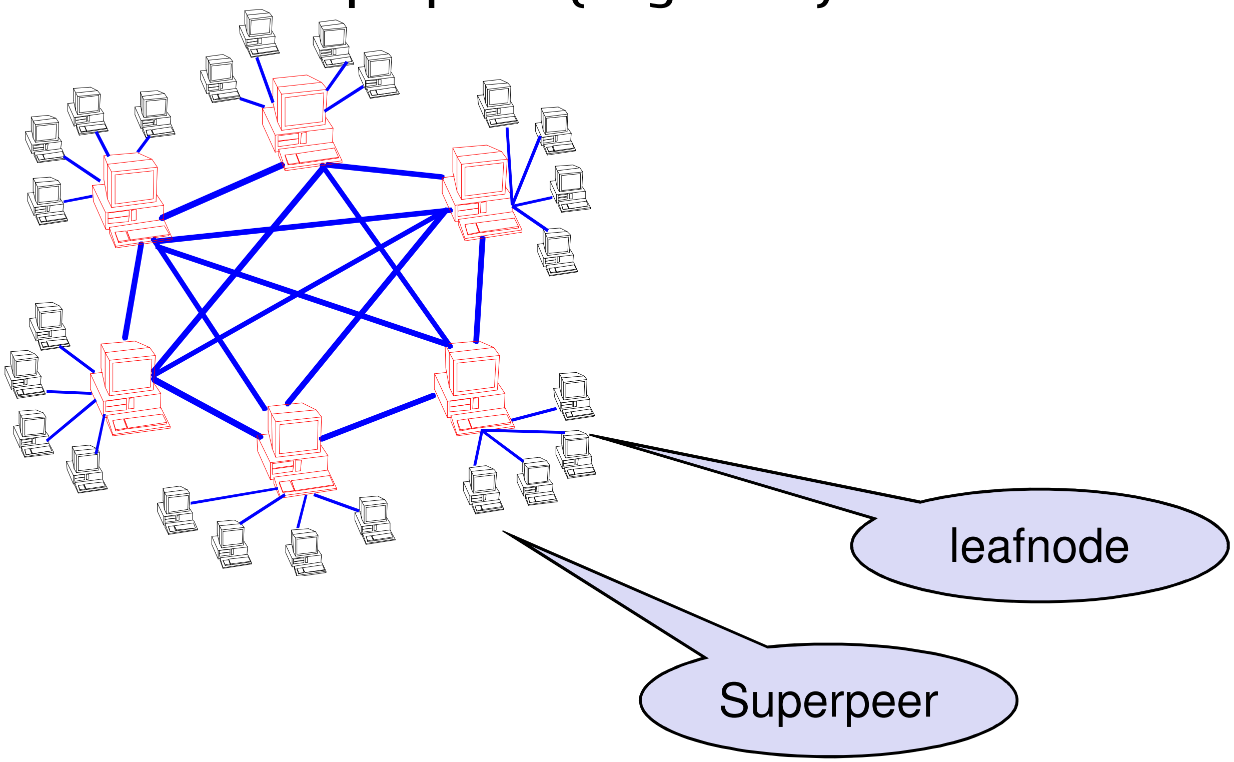
X. Overview

1. Centralized Peer-to-Peer Networks
2. Pure Peer-to-Peer Networks
3. Hybrid Peer-to-Peer Networks



X.4 Definition of Hybrid P2P

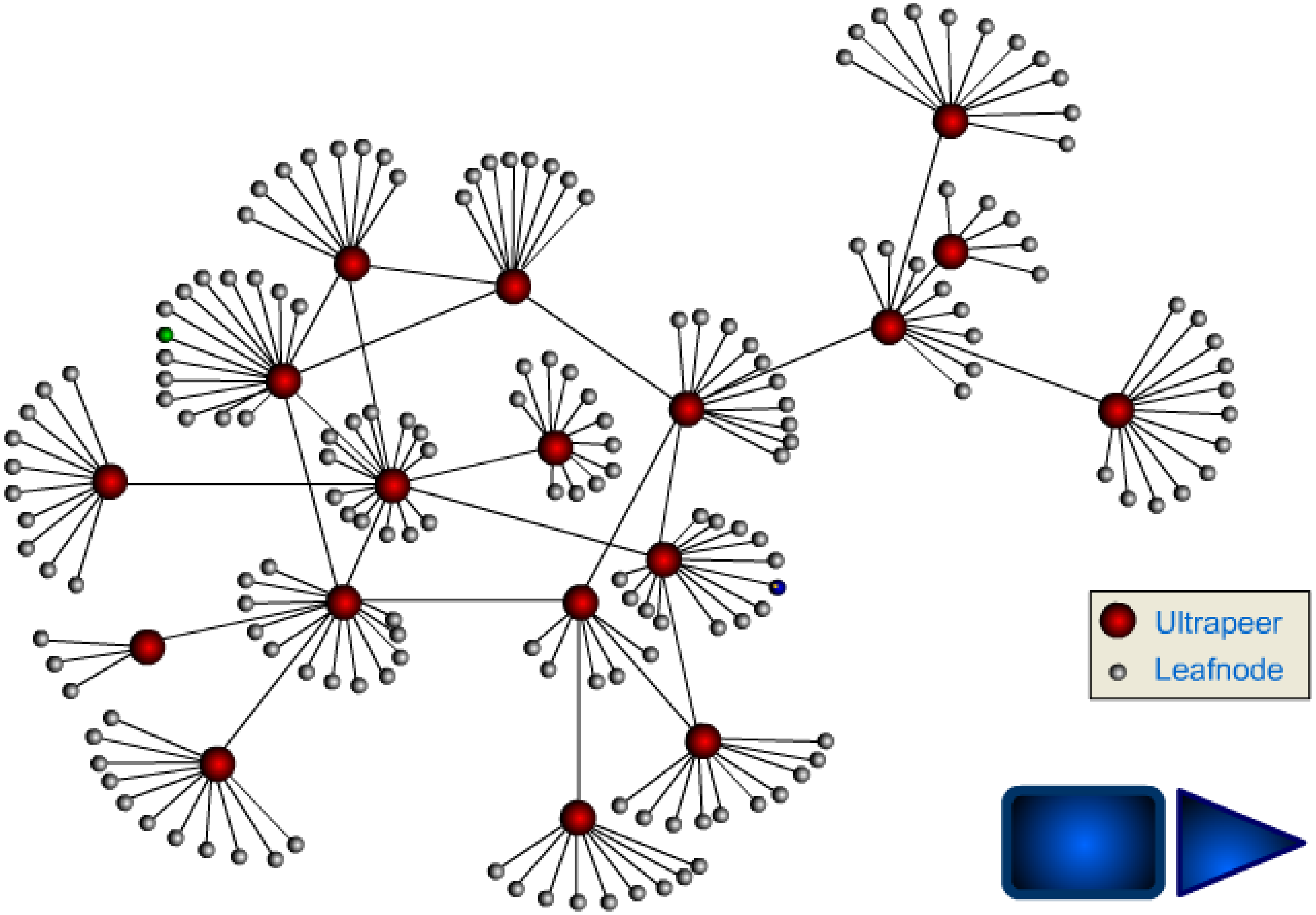
- Main characteristic, compared to pure P2P: Introduction of another dynamic hierarchical layer
- Hub based network
- Reduces the signaling load without reducing the reliability
- Election process to select an assign Superpeers
- Superpeers: high degree (degree>>20, depending on network size)
- Leafnodes: connected to one or more Superpeers (degree<7)



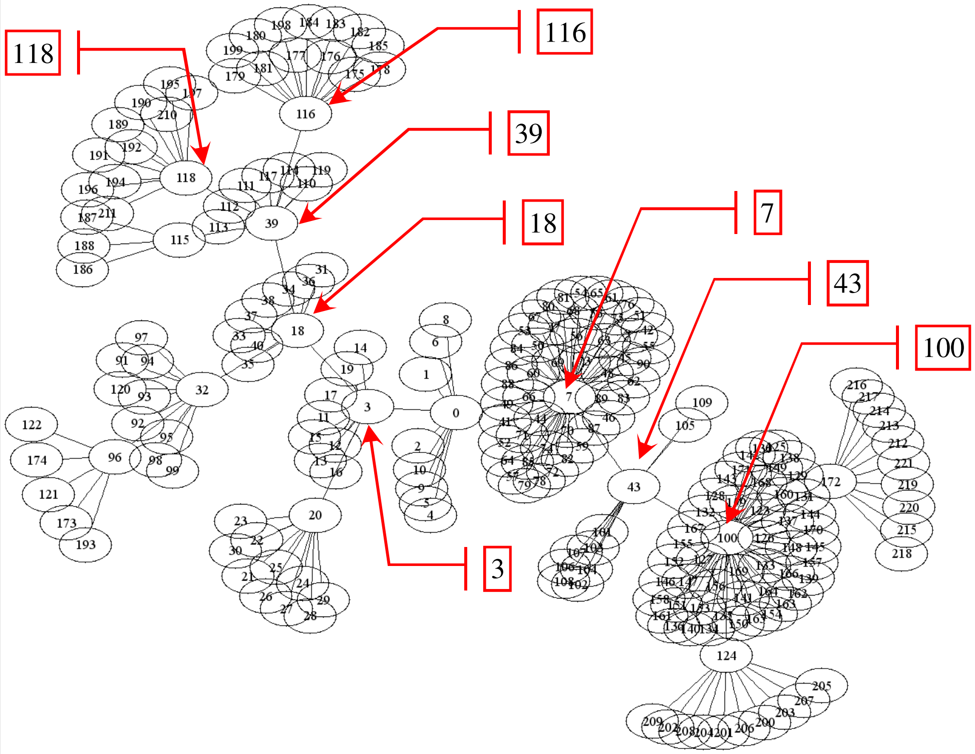
X.4 Example: Gnutella 0.6

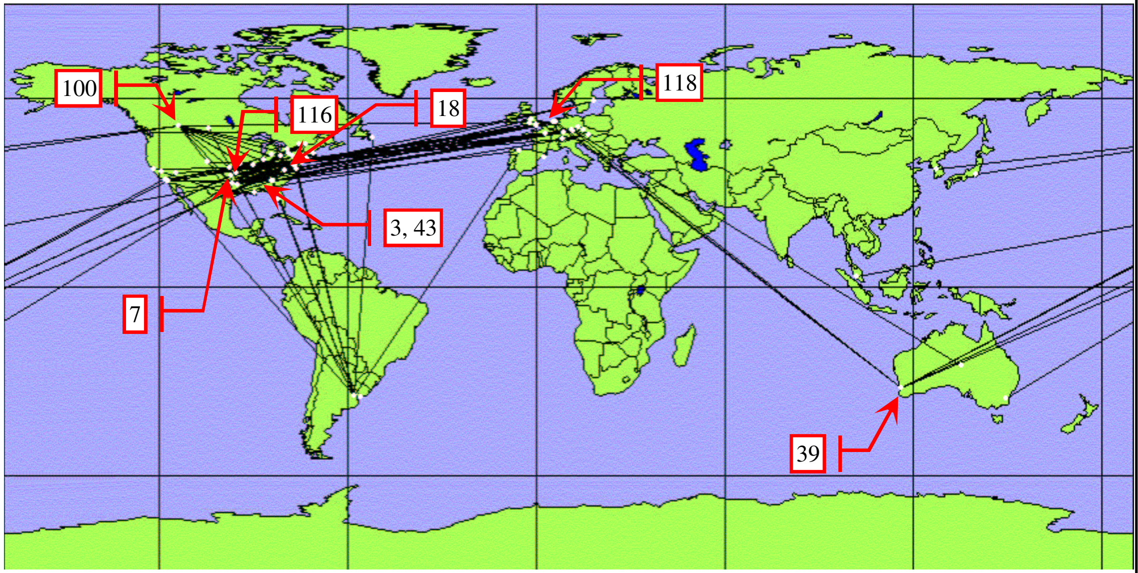
- Program for sharing files over the Internet
- Focus:
    - decentralized method of searching for files
    - Higher signaling efficiency than Pure P2P
    - Same reliability (no single point of failure)
- Basis of most file-sharing applications (not BitTorrent)
- Brief History:
    - **Spring 2001**: resulted from Gnutella 0.4 by further developments to improve scalability → Gnutella 0.6 (Hybrid P2P)
    - Since then:
        - available in a lot of implementations (Limewire, bearshare,...)
        - Developed further on (privacy, scalability, performance,...)

Gnutella 0.6 Animation



X.4 Topology of Hybrid P2P





Abstract network structure of a part of the Gnutella network (222 nodes Geographical view given by Figure on the right, measured on 01.08.2002

Geographical view of a part of the Gnutella network (222 nodes); The numbers depict the node numbers from the abstract view (Figure on the left, measured on 01.08.2002)

- Virtual network not matched to physical network. See path from node 118 to node 18.
- Superpeer (hub) structure clearly visible in abstract view

How Skype works

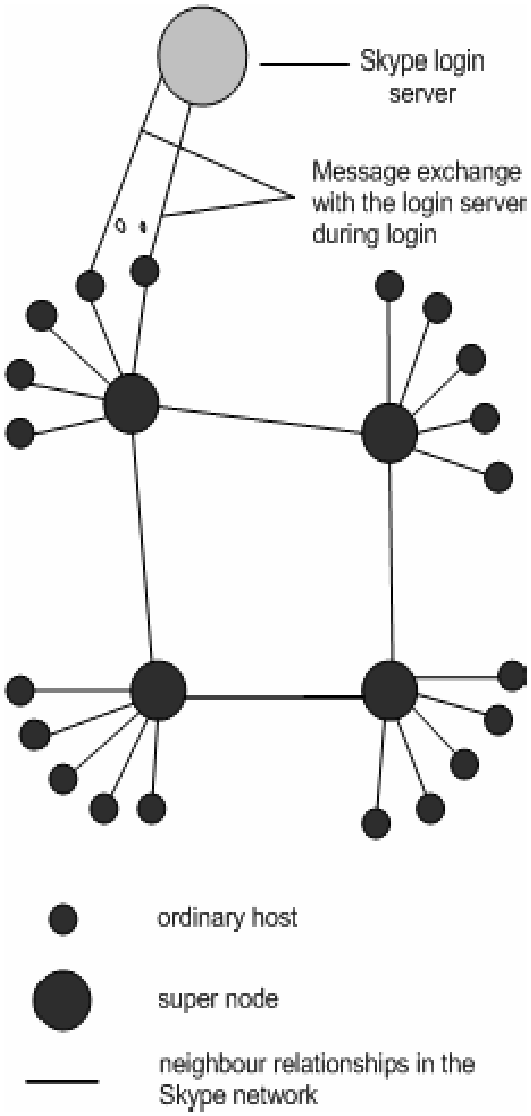

Skype login server
Message exchange with the login server during login
ordinary host
super node
neighbour relationships in the Skype network

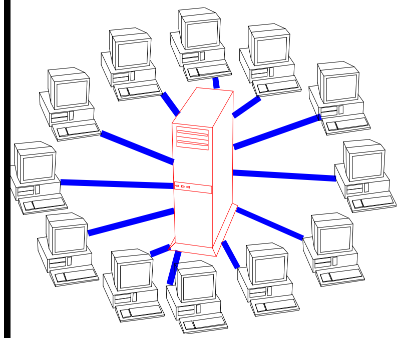
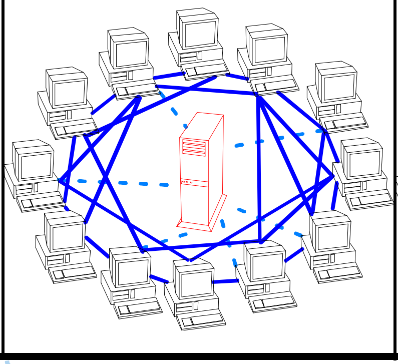
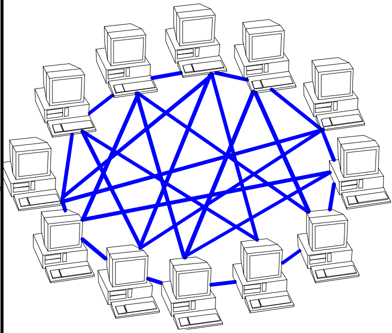
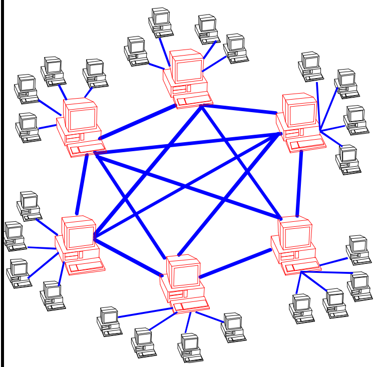
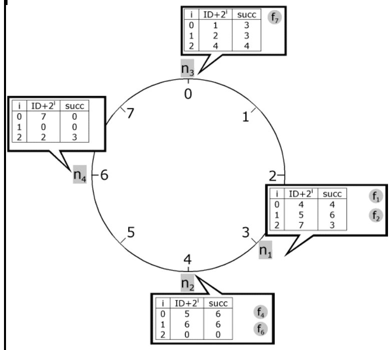
Figure 1. Skype Network. There are three main entities: supernodes, ordinary nodes, and the login server.

Salman A. Baset and Henning Schulzrinne. An analysis of the Skype peer-to-peer Internet Telephony Protocol, Columbia University

X.4 Discussion

- Advantages
  - No single point of failure
  - Can provide anonymity
  - Heterogeneous devices
- Disadvantages
  - Still high signaling traffic, because of decentralization
  - No definitive statement possible if content is not available or not found
  - Overlay topology not optimal, as
    - no complete view available,
    - no coordinator
    - Zigzag routes, loops
    - Can not be adapted to physical network completely because of hub structure
  - Asymmetric load (Superpeers have to bear a significantly higher load)
- Application areas
  - File-sharing (Edonkey, Kazaa/FastTrack, Emule)
  - VoIP (Skype)

X.1 The Architectures of 1st and 2nd Gen. P2P

Client-Server	***Peer-to-Peer***			
1. Server is the central entity and only provider of service and content. → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system Example: WWW	1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servent concept)			
	Unstructured P2P			***Structured P2P***
	Centralized P2P	*Pure P2P*	*Hybrid P2P*	*DHT-Based*
	1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database Example: Napster	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities Examples: Gnutella 0.4, Freenet	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities Example: Gnutella 0.6, JXTA	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are "fixed" Examples: Chord, CAN
				 i ID+2i succ 0 1 3 1 2 3 2 4 4 f7 n3 0 1 i ID+2i succ 0 7 0 1 0 0 2 2 3 7 n4 6 2 i ID+2i succ 0 4 4 1 5 6 2 7 3 f1 f2 5 3 n1 4 n2 i ID+2i succ 0 5 6 1 6 6 2 0 0 f4 f6

1st Gen.

2nd Gen.