Peer-to-Peer and Cloud Computing

Prof. Dr. Jörg Hähner

Part 2:

Overview of
Peer-to-Peer systems

Content

- History of Peer-to-Peer Systems
- Driving Forces and Impact of Peer-to-Peer-Systems
- Characteristics of Peer-to-Peer Systems
- Fields Related to Peer-to-Peer Systems
- Applications of Peer-to-Peer Systems
- Classes of Peer-to-peer Systems and Network Topologies

ARPANet and Early Internet (1)

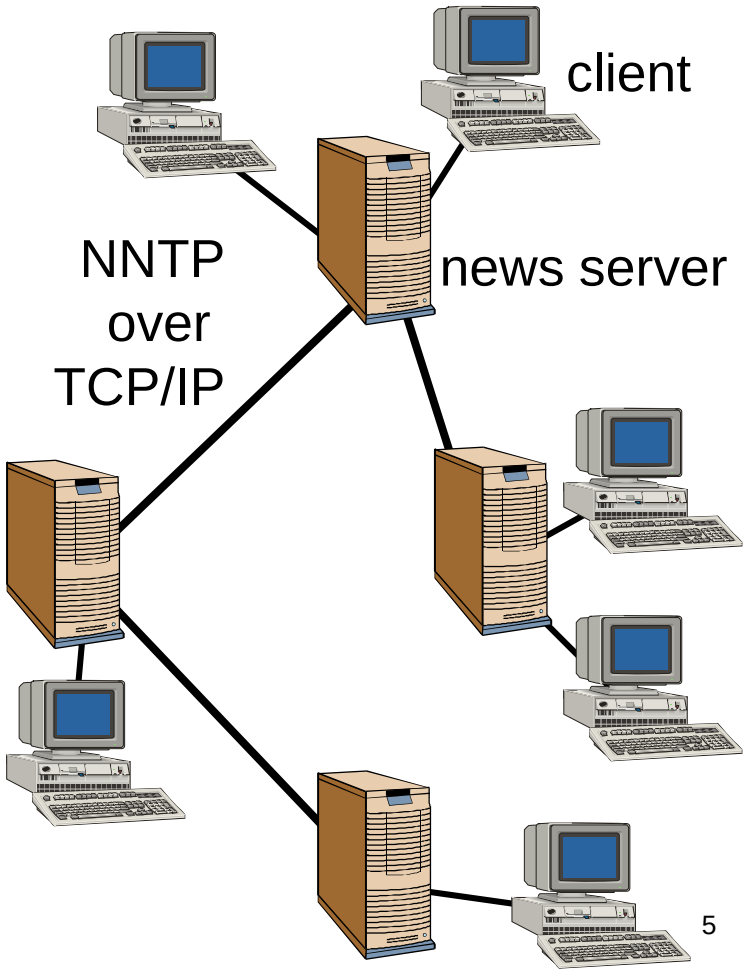
- 1967: **A**dvanced **R**esearch **P**rojects **A**gency (ARPA) of the U.S. Department of Defense initiates construction of computer network
- 1969: ARPANet connects four hosts of UCLA, SRI, UCSB, and the University of Utah
  - Hosts connected through Interface Message Processors (IMP)
  - Packet switched network
- 1973: ARPA starts “Internetting project”
  - TCP/IP becomes standard Internet protocol
- In 1980s:
  - Fast growth of network
  - Interconnected networks (e.g. ARPANet, NSFNet)

Goal: share computing resources and enable communication

- Community of peers: universities, scientists
  - No fixed provider/consumer relationship
- Typical applications: File transfer (FTP), Telnet, E-Mail, News (Usenet)
- E-Mail: Transfer messages between machines of equal users (“peers”)
- FTP & Telnet
  - Client/Server protocols
    - Client downloads files from server or logs into server
  - Usage pattern often was Peer-to-Peer
    - Hosts implemented clients **and** servers and were used as such
    - Static addresses, no firewalls, no NAT to prevent from connecting to hosts

Usenet: Bulletin system developed 1979

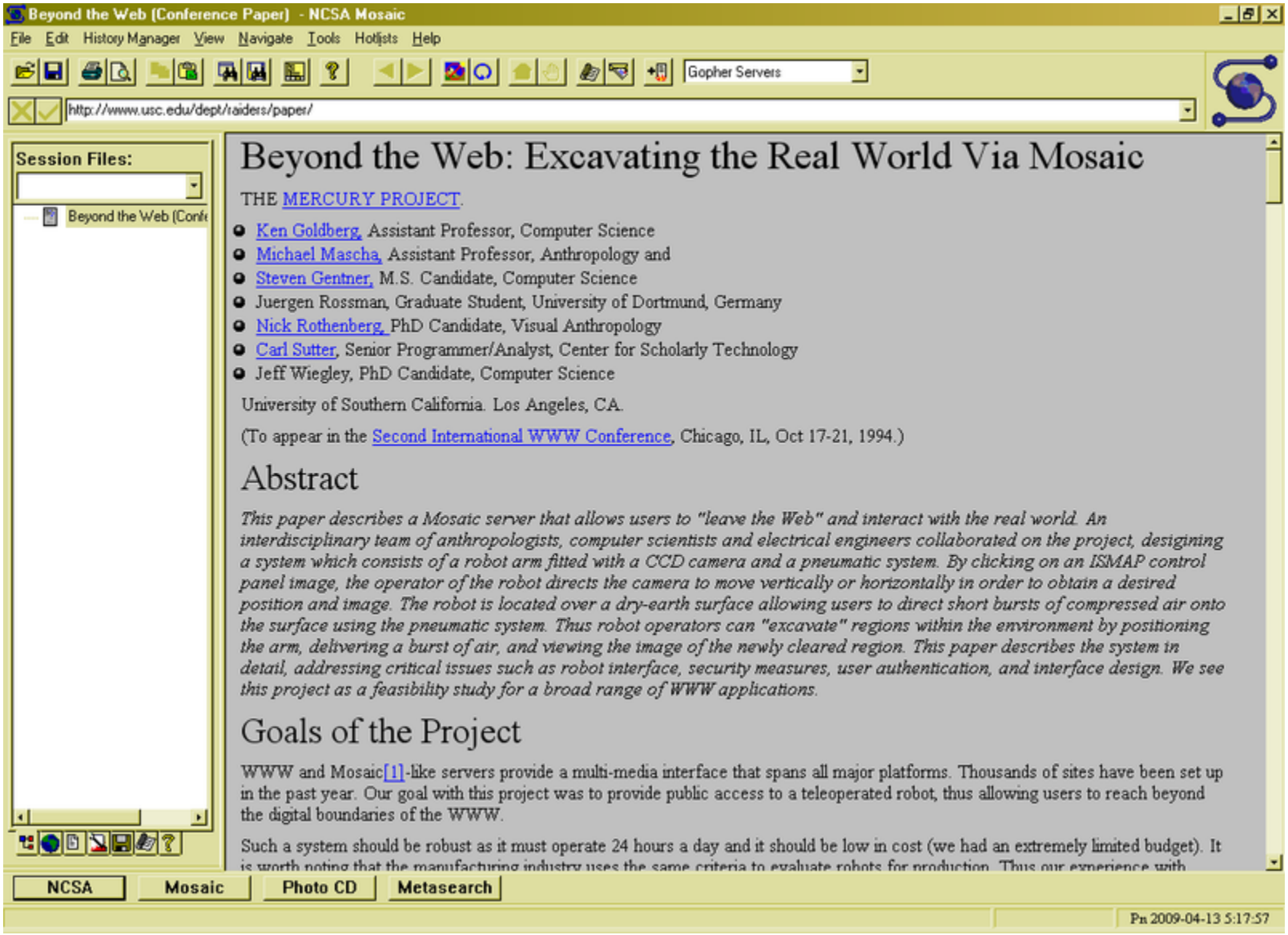
- Users post articles to news groups (comp.hardware, rec.music, etc.)
- Decentralised system
  - Network of news servers for storing and forwarding messages
  - News servers exchange messages via UUCP or NNTP
  - Operator autonomously decides which groups to offer
- News servers act as “peers“
- Client/server relationship between clients and news servers



Commercial Internet, World Wide Web, Microcomputers

- Late 1980s:
  - First commercial Internet service providers
  - Microcomputers start to replace minicomputers
- 1989: start of Web technology by CERN
- 1991: NSFNet opened for commercial applications
- 1993: “Mosaic“ browser
- 1995: Commercial providers now provide backbone of the Internet
- WWW becomes the killer application of the Internet

Mosaic Browser



Shift from peers to asymmetric communication

- Servers:
  - Expensive & well-connected (LAN & WAN)
  - Large storage, memory, processing power
- PCs:
  - Cheap
  - Small storage, memory, processing power
  - Good for applications like word processing, spread sheets, web browser, etc.
  - Connected to host through fast LAN or slow modem connections over PPP/SLIP

Few servers serve many PCs (clients): client/server paradigm

End of open Internet

- Large host number cannot be provided with static IP addresses
  - Dynamic IP addresses
    - Many users log in and out sporadically
    - IP addresses assigned dynamically
  - Network address translation (NAT): Hosts of one organisation get *one* public IP address to connect to Internet
- Firewalls
  - Public Internet leads to security risks
  - Clients are blocked from receiving incoming messages

These (useful) techniques prevent clients from becoming servers.

Peer-to-Peer Era

- Resources of PCs increase more and more
  - Storage, processing power, bandwidth
  - Bandwidth becomes cheap (flat rates)
- User PCs can be employed as servers
- 1999: Napster starts files sharing service employing user hardware
  - Service is very successful
  - Spawns lot of similar projects

Resources are shared between peers:

Peer-to-Peer paradigm

- Already ARPANet enabled sharing of resources between peers
- Some old technologies like USENET use decentralised approaches similar to current Peer-to-Peer applications

What‘s new about current Peer-to-Peer systems?

- Scale:
  - Millions of peers
  - Huge resources: for instance, large number of shared files
- Dynamics of resources:
  - Autonomous peers frequently join and leave the system

How to make such a system efficient, scalable, and manageable?

Content

- History of Peer-to-Peer Systems
- Driving Forces and Impact of Peer-to-Peer-Systems
- Characteristics of Peer-to-Peer Systems
- Fields Related to Peer-to-Peer Systems
- Applications of Peer-to-Peer Systems
- Classes of Peer-to-peer Systems and Network Topologies

Driving forces behind P2P: Terminals

- 1992:
  - Average hard disk size: ~ 0.3 GByte
  - Average processing power (clock frequency) of personal computers: ~ 100 MHz
- 2002:
  - Average hard disk size: 100 GByte
- 2004:
  - Average processing power (clock frequency) of personal computers: ~ 3 GHz

Personal computers today have storage capacities and processing power comparable to former servers!

Driving forces behind P2P: Networks

- Early 1990s:
  - private users connect to the Internet via 56 kbps modem connections
- 1997/1998:
  - first broadband connections for residential users become available
  - cable modem with up to 10 Mbps
- 1999
  - Introduction of DSL and ADSL connections
- Today
  - Data rates of more than 16 Mbps via common telephone connections become available.
  - Deregulation of the telephone market leads to cheap "flat rates".

Today, bandwidth is plentiful and cheap also for private users!

Driving forces behind P2P: Pragmatism (1)

Implement desired functionality without …

- … lengthy standardisation
- … support of provider
- … control of central authority (single point of failure)

- Example 1: P2P telephony / instant messaging (e.g. Skype, until Microsoft bought them)
  - Implemented without support of carrier
  - Functionality can be extended immediately without standardisation by ITU, IETF, … (in contrast to SIP)

- Example 2: Multicast
  - Slow take-off of IP multicast
  - Multicast protocols implemented on the application layer/end system between peers
- Example 3: The Napster story (part 1)
  - Napster system: P2P file sharing
  - MP3 files stored on peers
    - Large storage and bandwidth requirements
      - Central file server and bandwidth too expensive for company
    - Copyrighted material → central server could easily be shutdown on demand of record companies
      - Peers are autonomous and large number of owners is hard to tackle

- Example 3: The Napster story (part 2)
  - Napster system:
    - Central lookup service → single point of failure
  - Lawsuit against Napster
    - Napster sets up filters (which were ineffective)
  - July 2001: Napster Inc. is convicted
    → shutdown of central servers
  - Other fully decentralised P2P systems become popular, e.g., Gnutella
    - Censorship not easily possible in fully decentralised Peer-to-Peers systems

Large amounts of P2P-based Internet traffic:

- Up to 60% in German research network DFN in 2002 [DFN Mitteilung 3/2002]
- 60% of Internet traffic in 2005 [CacheLogic]
- 30% of Internet traffic during day, 70% during night in 2006 [ipoque GmbH]

Growing number of successful P2P applications

- Community projects
- Commercial applications

First de-facto standards for P2P application development

- Frameworks & platforms supported by companies

P2P has been termed a disruptive technology that could replace common client/server systems

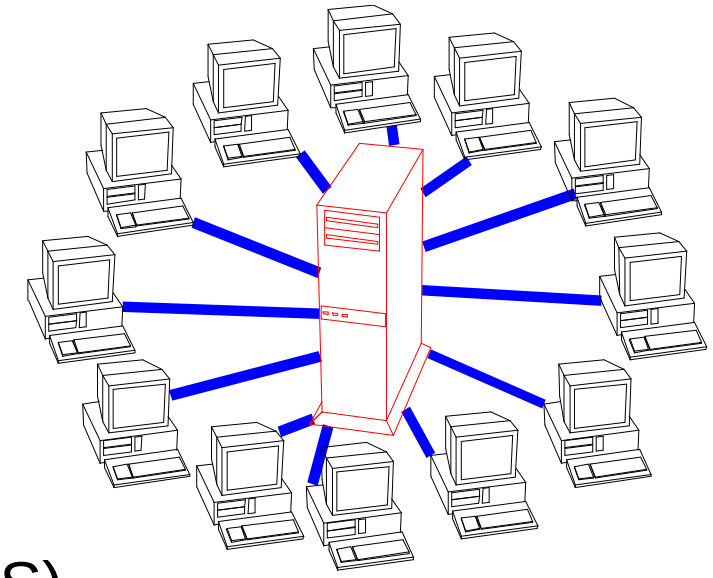
- Still there are successful client/server systems like Flickr or YouTube (also provide large resources for free)
- Traditional client/server systems might apply concepts from P2P systems to make them, e.g.:
  - more robust
  - scalable

- History of Peer-to-Peer Systems
- Driving Forces and Impact of Peer-to-Peer-Systems
- Characteristics of Peer-to-Peer Systems
- Fields Related to Peer-to-Peer Systems
- Applications of Peer-to-Peer Systems
- Classes of Peer-to-peer Systems and Network Topologies

Client / Server Paradigm

- Dedicated servers provide service
- Clients use service

→ Clear separation of functionality, i.e., asymmetric functionality



- Example: File service (Samba, NFS)
  - File servers provide storage accessible through network connections
  - Clients (workstations, laptops, etc.) access files on server

Pros (advantages of C/S systems):

- Scalability of *distributed* C/S systems
- Ease of administration
  - Well-known resources are easily accessible by provider
- E.g. easy implementation of access control or accounting
  - Operator has full control of server
- Easy prediction and assurance of Quality of Service (QoS)
  - Well-known resources; no/small resource fluctuation
- Resource can be found easily since it is at a well-known location
- Standardised interoperability

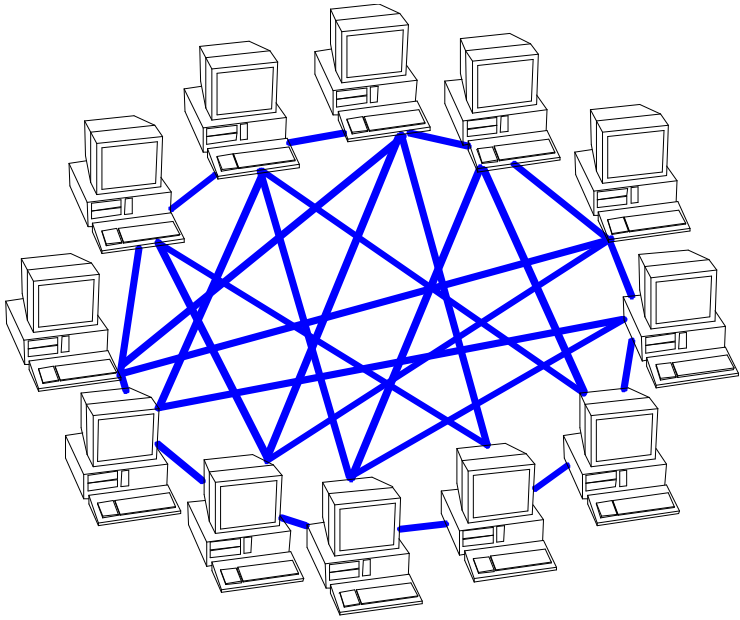
Cons (disadvantages of C/S systems):

- Scales only to a certain point
  - A distributed server (server farm, cluster) can be very powerful!
  - Scaling beyond this point might be very expensive
- High cost of ownership
  - High-end servers too expensive for small companies and communities
- Not as robust as a highly distributed P2P system
  - However: also a distributed C/S system can be very robust!
- Privacy
- No autonomy of users

Peer-to-Peer Paradigm

- Network of equal peers
- Peers use service and participate in providing the service

→ Peer is servent, i.e.,
server *and* client



- Example: File sharing (e.g., Gnutella)
  - Network of PCs → Peers
  - Peers downloads files (client role)
  - Peers provides local files to others (server role)
  - Peers forward requests for files they do not provide themselves

Different definitions in the literature:

"Peer-to-Peer is a class of applications that take advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet."

[C. Shirky: *What is p2p … and what isn't*. Network, O'Reilly, 2000]

Peer-to-Peer Systems: Definition(s)

"Peer-to-Peer systems are distributed systems consisting of interconnected nodes able to self-organise into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance, without requiring the intermediation or support of a global centralised server authority."

[S. Androutsellis-Theotokis, Diomidis Spinellius: A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys, 36(4), 2004]

Peer-to-Peer Systems: Definition(s)

"A Peer-to-Peer system is a self-organising system of equal, autonomous entities (peers) which aims for the shared usage of distributed resources in a networked environment avoiding centralised servers."

[R. Steinmetz and K. Wehrle: *Peer-to-Peer networking and computing*. Informatik-Spektrum, 27(1), 2004]

Peers:

- Equal: symmetric functionality, rights, and duties
  - Each peer is client and server, i.e., peers are servents
    - Peers use shared resources of other peers (client functionality)
    - Peers provide resources to other peers (server functionality)
  - However: some peers may provide more resources
    - Every peer should provide its share to keep the system running
    - “Free-riders” (pure clients) are no peers
  - Peers may be specialised to improve system performance
    - More powerful peers may implement special functionality (e.g. for resource lookup) → super peers
- Autonomous: not controlled by other entities
  - Peer has full control of its resources
  - Peer may join or leave the network at any time

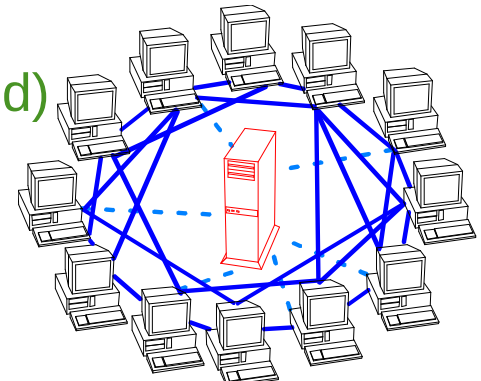
Resources:

- Shared: peers add their resources to the system
  - Typical resources:
    - Content
    - Storage
    - Computational power
    - Bandwidth
  - Each peer provides its share of resources to the system
  - Idea: more users → more resources

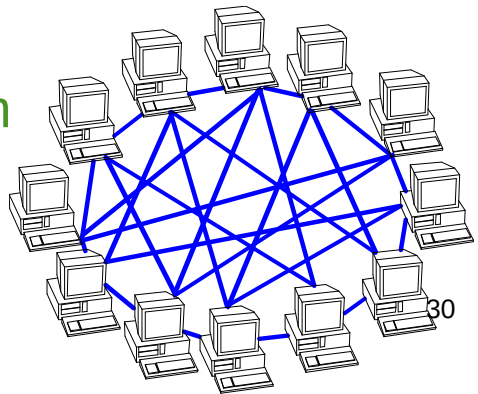
Resources (continued):

- Distributed:
  - Resources: at the “edge” of the network rather than the center
  - Peers connected by network for direct P2P resource access
  - No central server providing resource
    - No single point of failure with respect to shared resource
  - However: system may contain centralised functionality
    - Pure P2P system: no central component/single point of failure
    - Hybrid / centralised P2P system contains server component

Hybrid (centralised) P2P system



Pure P2P system



Note: variations from this basic principle exist!

Self-organising system (the basic principle)

- No central control or coordination of resources
  - Peers access other peers' resources without coordination by central server
- Structure of P2P network emerges through distributed decisions of peers
  - Peers autonomously build P2P network
- System constantly adapts to dynamic resources
  - Peers joining, leaving, failing

Coping with a large number of autonomous and dynamic peers is a key distinction of a P2P system to other distributed systems.

Paradigm shift in Internet communication:

Client/Server	→	Peer-to-Peer
coordination	→	cooperation
centralisation	→	distribution
control	→	incentives

Pros (advantages) of P2P systems

- Scalability through resource aggregation
  - Incorporates vast peer resources
  - System grows naturally with numbers of peers (more users → more resources)
  - No resource bottlenecks at central components
- Cost sharing / reduction
  - Each peer adds its resource share to the system
- Robustness
  - Many redundant resources

Pros (advantages) of P2P systems (continued)

- Autonomy
  - No central control
- Anonymity and privacy
  - No central instance (server) that knows everyone
- Dynamism and ease of administration
  - Self-organisation
  - Capability to deal with large numbers of dynamic and unreliable resources

Cons (disadvantages) of P2P systems:

- Access control, accounting, etc. difficult
  - Peers are distributed and autonomous
- Quality of service hard to predict and assure
  - High resource fluctuation
  - Mostly best-effort service
- Resource cannot be found easily
  - Lookup mechanism required

Content

- History of Peer-to-Peer Systems
- Driving Forces and Impact of Peer-to-Peer-Systems
- Characteristics of Peer-to-Peer Systems
- Fields Related to Peer-to-Peer Systems
- Applications of Peer-to-Peer Systems
- Classes of Peer-to-peer Systems and Network Topologies

Key concepts of a distributed system:

- Multiple nodes working independently of each other. (Where a node consists of at least one CPU with associated memory.)
- The nodes have no common shared memory.
- There exists an interconnection network.
- There is no *Single Point of Failure*.

Distribution transparency (a distributed system looks like a centralised system from outside):

- Access transparency: Local and remote resources are accessed in the same way.
- Location transparency: The location of objects is not known to the users.
- Replication transparency: The number of copies that exist of an object is not known to the users.
- Fragmentation transparency: Objects are accessed without knowledge about any possible fragmentation.

→ Pure P2P systems are distributed systems!

Key concepts of a grid system:

- Coordination of distributed resources that are not subject to centralised control
  - Integration of resources from different administrative units (different units of one or different companies/organisations)
- Using standard, open, general-purpose protocols and interfaces
  - Authentication / authorisation
  - Resource discovery / resource access
  - …

Key concepts of a grid system (continued):

- Deliver nontrivial qualities of service to the user
  - Response time
  - Throughput
  - Availability
  - ...

Grid

- Community: companies and research institutes
  - Interests: quality of service, accounting, authorisation
- Resources: powerful, well-connected, (clusters, pools of workstations), always connected

Peer-to-Peer

- Community: diverse individuals
  - Interests: best-effort, anonymity
- Resources: consumer hardware, consumer network, (PCs, DSL connection), intermittent network connection

Comparison: Grid & Peer-to-Peer Systems (2)

Grid

- Scale: Thousands
- Infrastructures: Standardised, multi-purpose (authentication, authorisation, resource discovery, data management)

P2P

- Scale: Millions
- Infrastructures: Protocols for specific functionality (e.g., routing and object localisation)

There is no hard line between both fields, but a different focus. Many P2P concepts are applicable to grid and vice versa.

Content

- History of Peer-to-Peer Systems
- Driving Forces and Impact of Peer-to-Peer-Systems
- Characteristics of Peer-to-Peer Systems
- Fields Related to Peer-to-Peer Systems
- Applications of Peer-to-Peer Systems
- Classes of Peer-to-peer Systems and Network Topologies

Structured according to the **primary** shared resource:

- Files / content
- Information
- Bandwidth
- Storage
- Processing power

Note: This is not exclusive!

All P2P systems share some bandwidth, storage, and processing power to keep the system running!

- Peers offer locally stored files to other peers
  - Music, videos, documents, etc.
- Most prominent P2P applications:
  - Napster, Gnutella, eDonkey, KaZaA, BitTorrent, ...
  - Substantial share of Internet traffic
- Often subject to legal issues
  - Centralised systems can be closed easily
  - Decentralised P2P systems hard to control

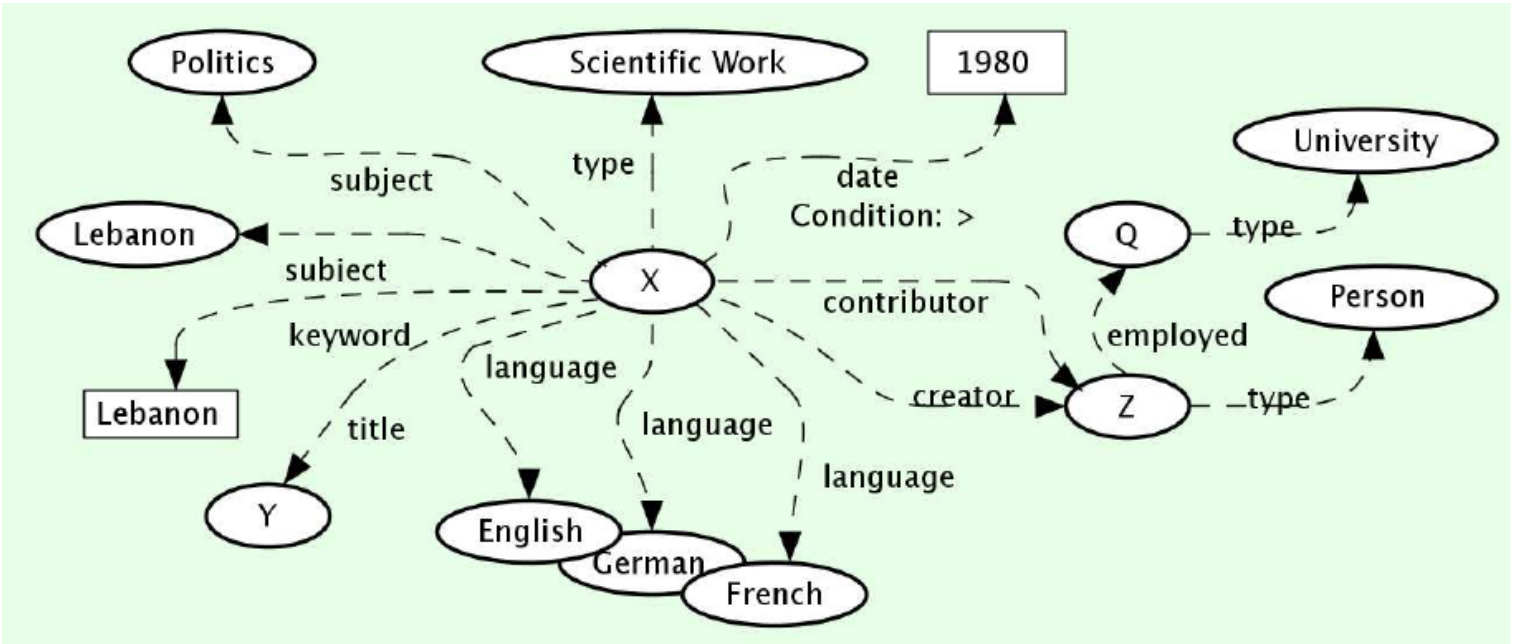
- Git (“Blödmann”)
  - is a distributed revision control
  - and source code management (SCM) system
  - with an emphasis on speed.
- Git was initially designed and developed by Linus Torvalds for Linux kernel development; it has since been adopted by many other projects.
- Basis is a P2P-similar approach for data sharing.
  - Each Git repository is simultaneously client and server.
  - Keeps track of its local revisions and exchanges data (information about revisions) with others.

Information (1)

- Peers provide information about objects to other peers
- Information structured according to some schema
- Peers run local databases / repositories
- Challenges:
  - Supporting complex queries on object attributes, e.g.:
    - Ranges
    - Multi-attributes
  - Integration of heterogeneous sources
    - Mapping between local schemas / ontologies

Example: Edutella

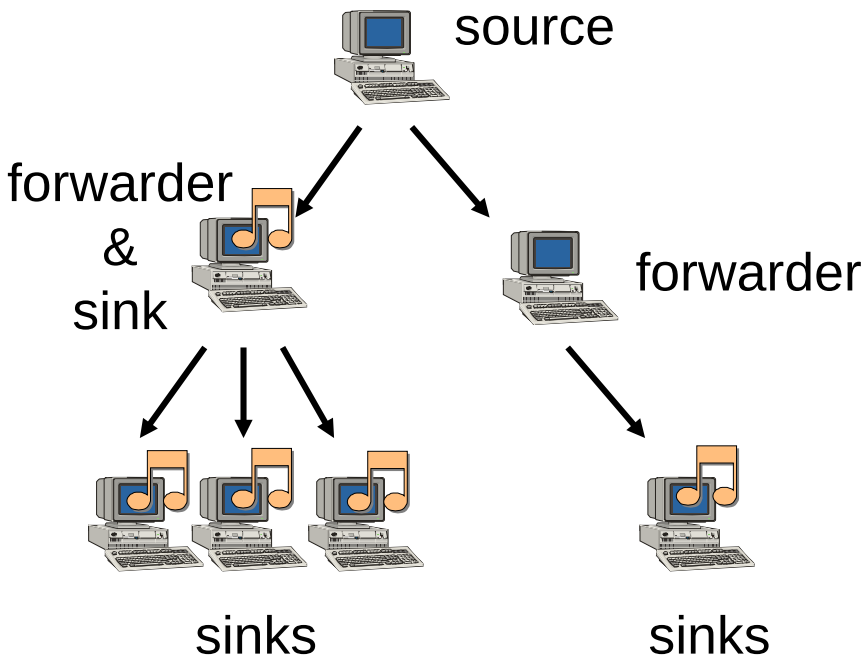
- Exchange of information about educational objects
- Query: scientific works on the subject of politics, having Lebanon as subject or keyword, with a title (Y), written in English, German or French, created or contributed to by a Person (Z), employed at a University, and created after 1980.



[Source: Mikael Nilsson: *The Edutella P2P Network. Supporting Democratic E-learning and Communities of Practice.*]

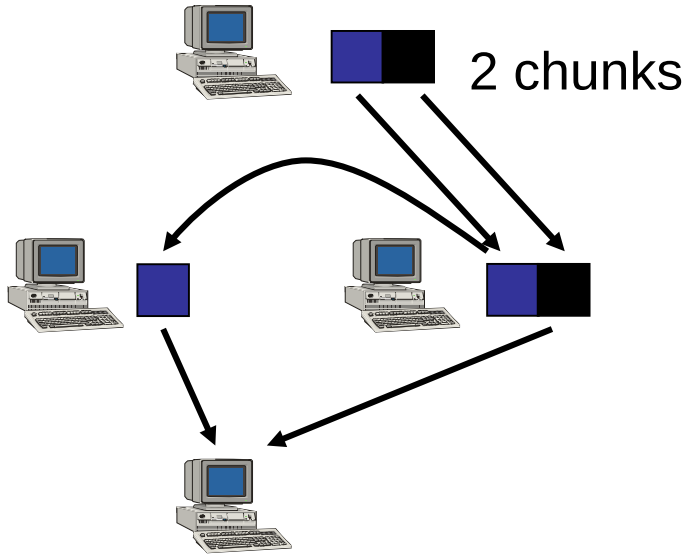
Bandwidth (1)

- Peers offer bandwidth for transmitting data to other peers
- Application-level multicast
  - Multicast on IP layer not well-supported by ISPs
  - Peers act as forwarders for data to other peers
    - Multicast tree built of peers
  - Example: PeerCast.org
    - P2P streaming of music and videos
    - Each peer is forwarded for a user-defined number of downstream peers



File swarming

- Initially, complete file is stored at one peer only
- File is cut into chunks
- Peers download chunks
- Peers offer completely downloaded chunks to other peers
- Bandwidth load is distributed among peers
- Example: BitTorrent



peer downloads chunks from peers that have already downloaded chunk

Peer-to-peer storage network:

- Peers offer their storage capacity to others (or pay a fee)
- Files or file parts are stored at different peers
- Challenges:
  - Availability and durability
    - Peers may leave the system unexpectedly!
    - Replication and caching required
  - Security
    - Peers may be untrusted
    - Encryption required
- Examples: Pasta, Oceanstore, MojoNation

Peers share their processing power with others

- Bundle peers to build a single logical computer
  - Provide processing power that is equivalent to a super computer
  - Early projects utilise spare processor cycles, e.g.:
    - Search for extraterrestrial intelligence (SETI@home)
    - Great Internet Mersenne Prime Search (GIMPS)
    - Genome@Home
  - Standardised middleware for grid applications: Globus Toolkit

Content

- History of Peer-to-Peer Systems
- Driving Forces and Impact of Peer-to-Peer-Systems
- Characteristics of Peer-to-Peer Systems
- Fields Related to Peer-to-Peer Systems
- Applications of Peer-to-Peer Systems
- Classes of Peer-to-peer Systems and Network Topologies

- P2P system may consist of a large number of dynamic peers sharing resources
- One peer cannot know all other peers and their resources (no global knowledge)
- Key question:
  How can a peer find peers providing certain resources and communicate efficiently with a set of peers?

→ Network of peers for resource lookup and routing

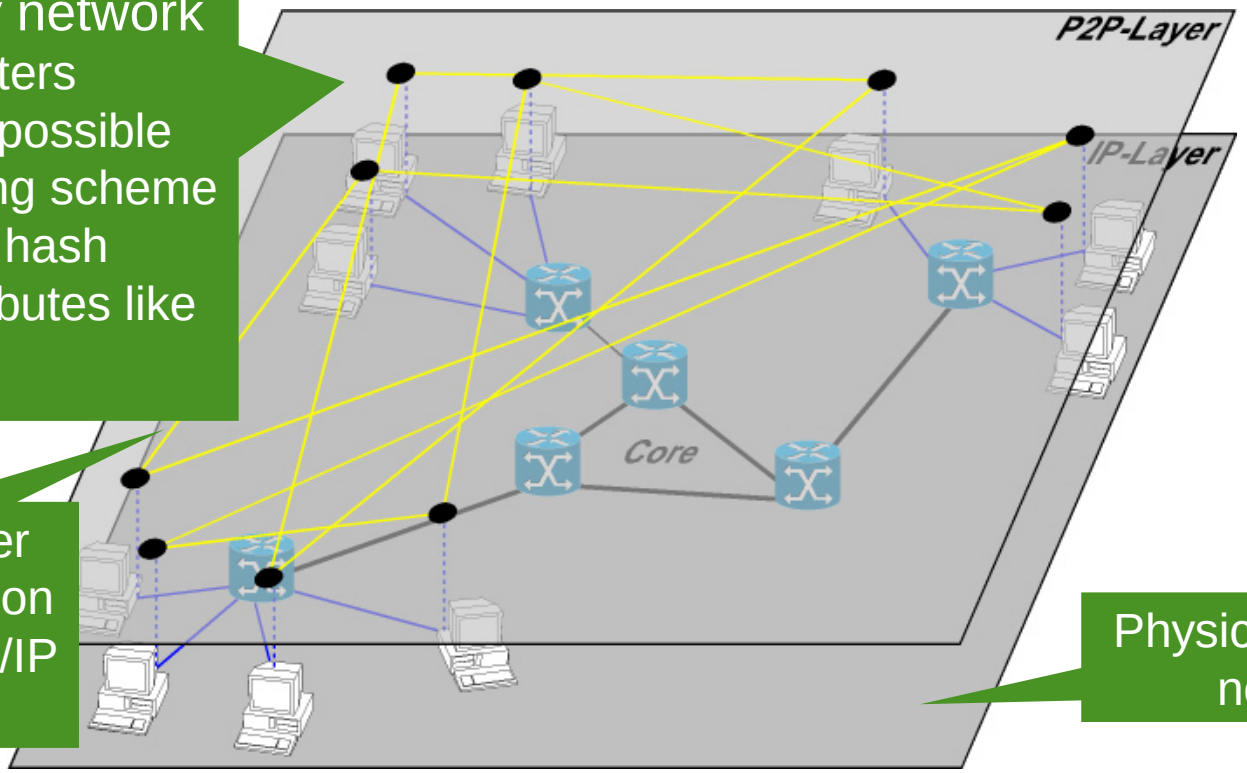
  – Network typically implemented by end systems on application layer → Overlay network

Logical network built on top of physical or another logical network: overlay on top of underlay network

- Logical links in overlay network are paths in underlay network
  - one hop in overlay → multiple hops in underlay
- Routing functionality implemented in underlay *and* overlay network
- Overlay can have any topology
  - Topology of overlay can be independent of underlay topology
  - Efficiency of overlay network can be increased by considering underlay topology
- Overlay network can use its own addressing scheme

Peers are organised in **overlay networks** on top of IP infrastructure

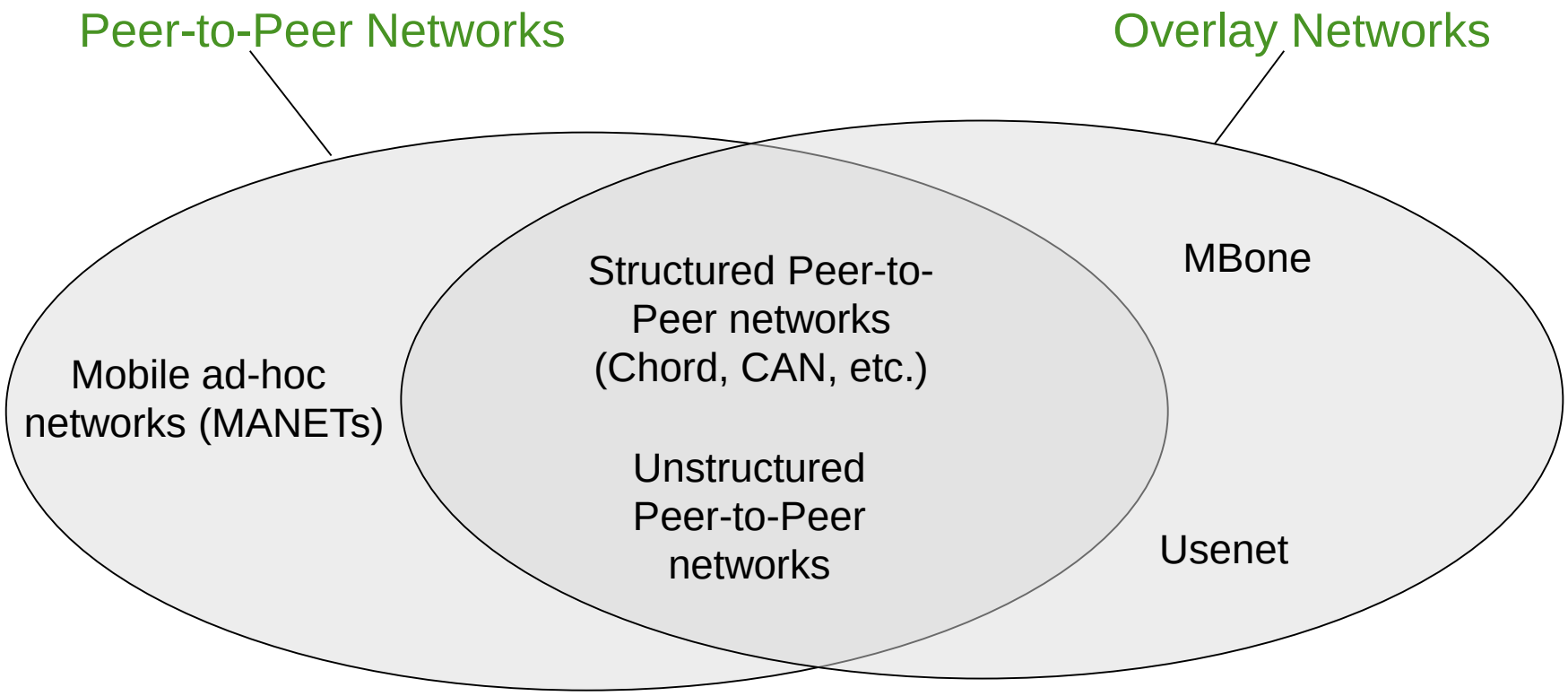
Logical overlay network
- Peers are routers
- Any topology possible
- Any addressing scheme possible (e.g. hash values or attributes like artist or title)



Peer-to-Peer communication through TCP/IP or UDP/IP

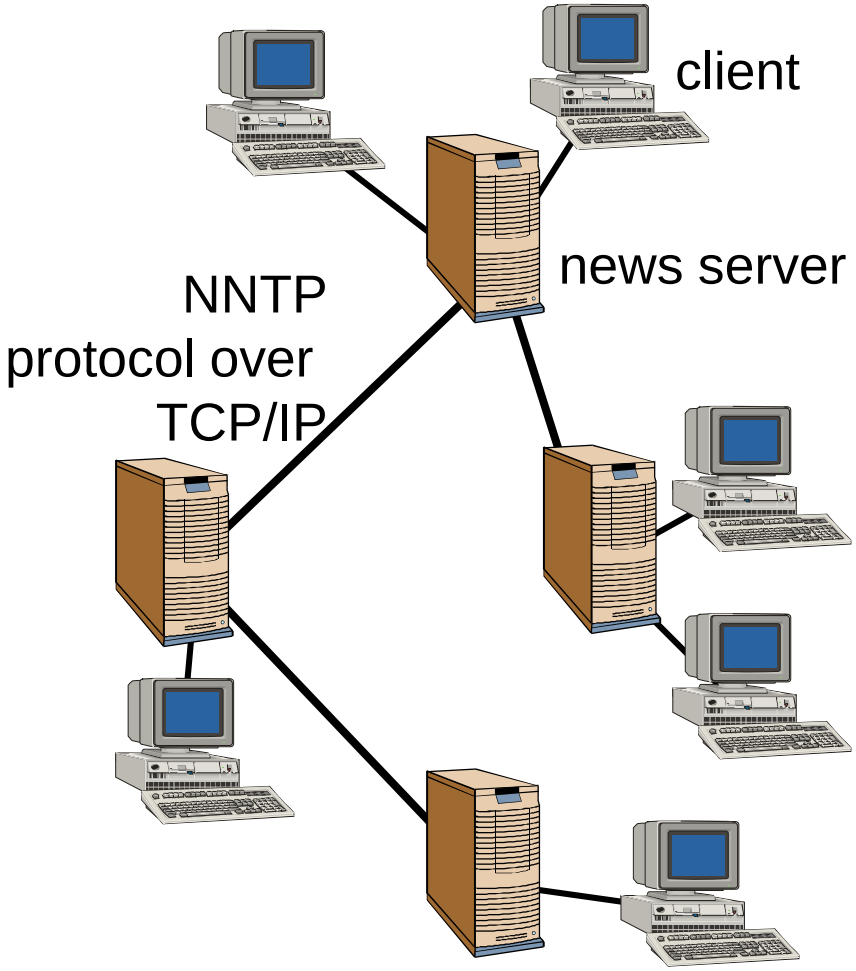
Physical underlay network

Overlay Networks and Peer-to-Peer Networks



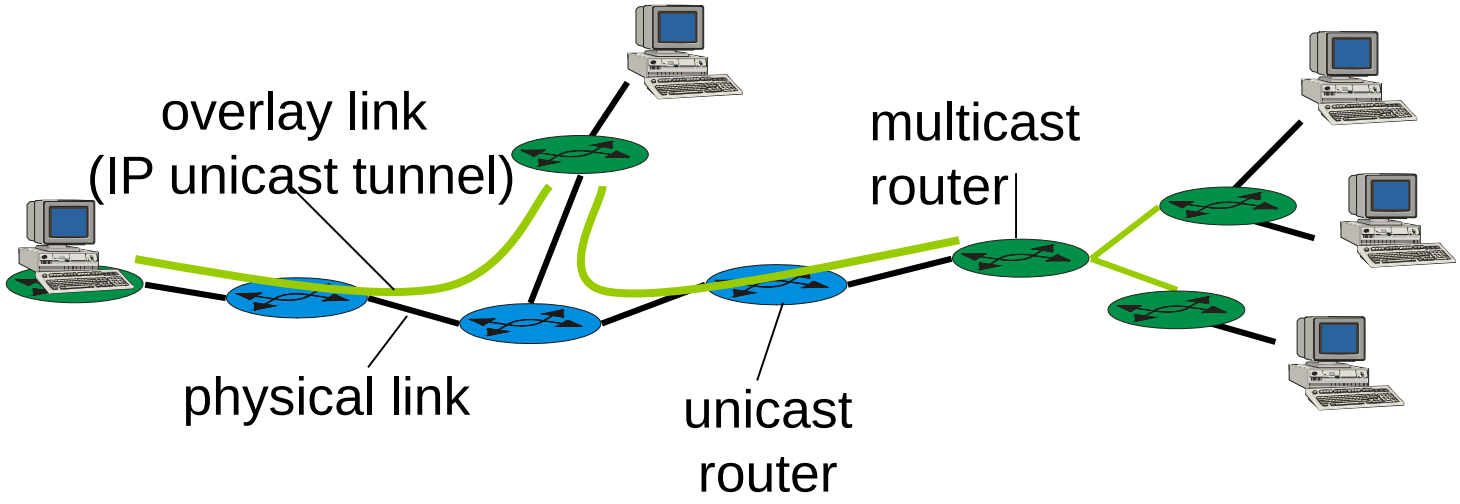
Traditional Overlay Networks: Usenet

- Clients post and fetch articles of news groups to/from news servers
- Servers store and forward messages
- Overlay network of servers
  - Servers pass new messages to neighbouring servers
    → flooding of overlay network
- Decentralised but not P2P:
  - No self-configuration: links defined manually
  - Fixed client/server functionality



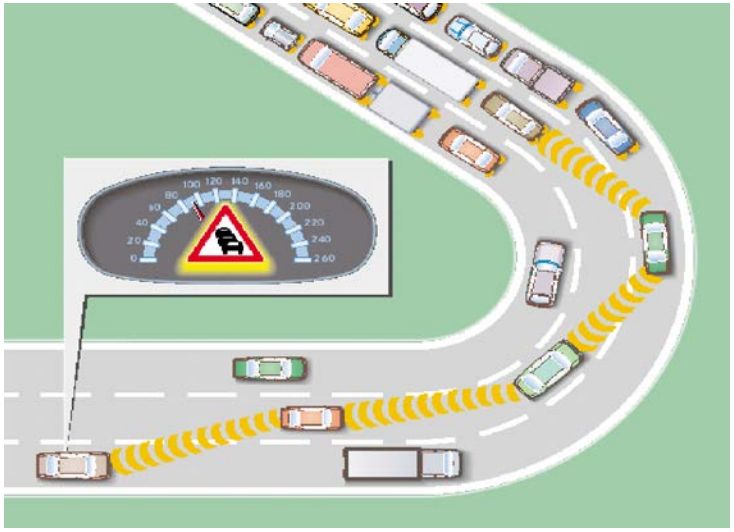
Traditional Overlay Networks: MBone

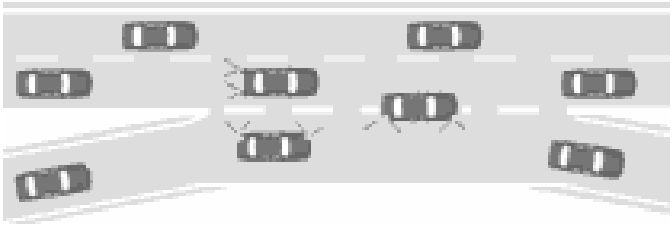
- Currently, many operators do not support IP multicast
- MBone: Multicast-aware overlay network
  - Multicast-capable “islands“ connected by unicast tunnels (IP in IP forwarding)
- Decentralised but not P2P:
  - No self-configuration: network structure defined manually
  - End systems may not forward messages (client/server-like)



Peer-to-Peer System Without Overlay: MANETs

- Data exchange between mobile devices





- Mobile devices organised in dynamic ad-hoc networks
  - Neighbours: devices within radio range (physical links)

  → No overlay network
- Equal peers: all devices may be message sources, sinks, or routers, network is self-organised

  → MANET is a P2P system

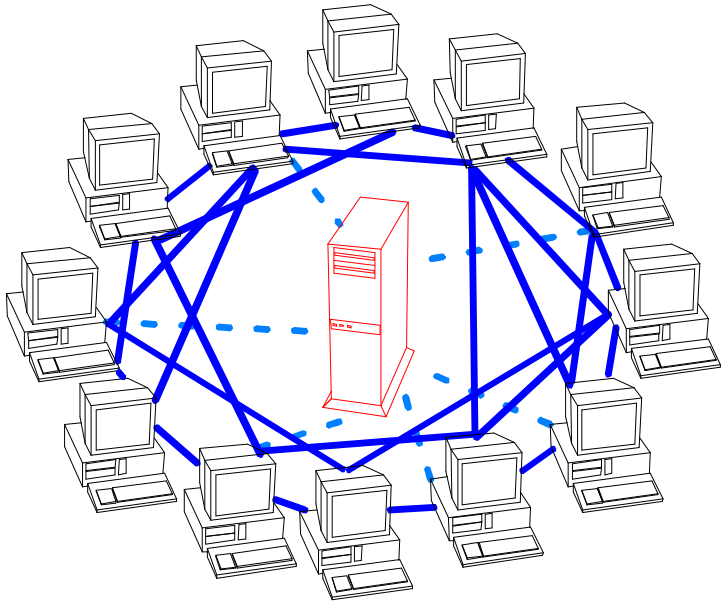
Special problems in MANETs:

- Mobility & wireless communication
- Discussed in lecture “Ad-hoc and Sensor Networks” (summer term)

- Creation of network topology non-deterministically as nodes join and leave the system
  - “Random” graph
  - Network topology unrelated to content provided by peers
- Only simple search mechanisms possible
  - “Undirected” query routing, heuristics
- Cope very well with transient node populations
  - Insertion and removal of peers is easy since no particular network structure has to be maintained.
- Examples: Gnutella, Kazaa, FreeHaven

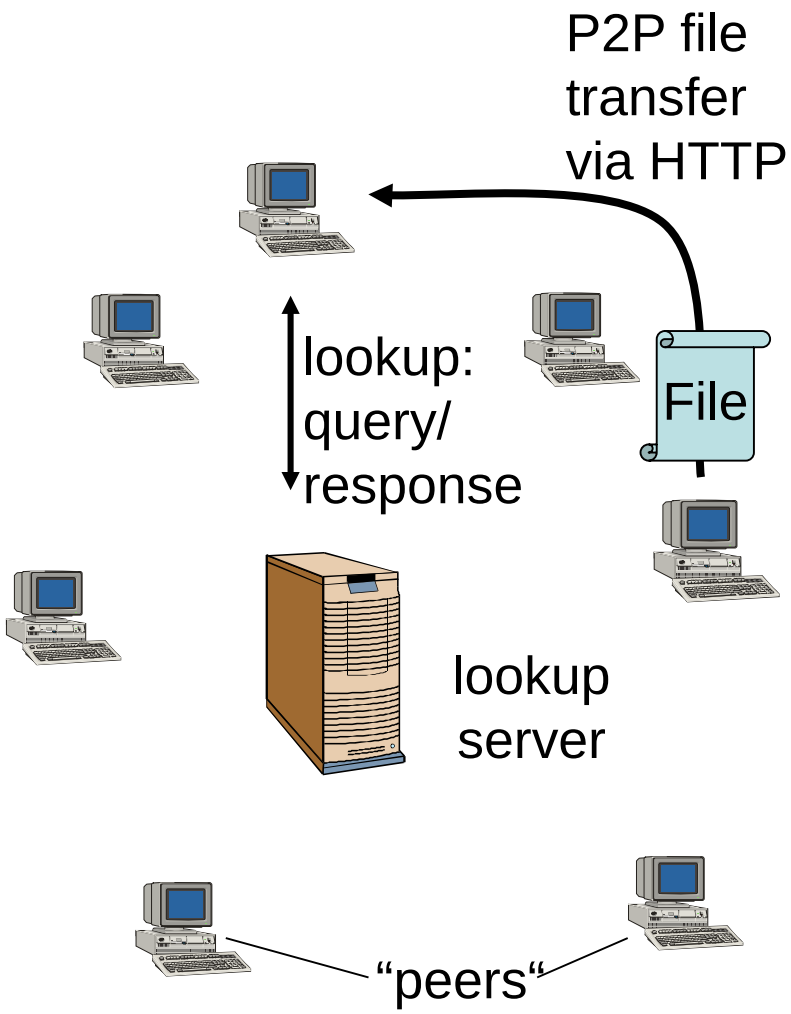
- Topology of overlay network tightly controlled
- Content(-pointers) placed at precisely specified locations/peers
  - Peers are responsible for certain keys
- Increased scalability for *exact-match* queries
  - Efficient key lookup through DHT (directed query routing)
  - Complex queries require special concepts
- Insertion and removal of peers is more complex
  - Topology of the overlay must be maintained
  - Content(-pointers) may have to be migrated on insertion/removal of peers
- Examples: Chord, Pastry, CAN

- Central server implements peer lookup
  - Server is index/group database
  - Client/server communication between peers (clients) and central server
  - Single point of failure!
- Shared and distributed resources are used in P2P manner
  - Peers directly access resources of other peers via P2P network
  - Peer (not server!) can be removed without loss of functionality
- Hybrid of client/server and P2P system

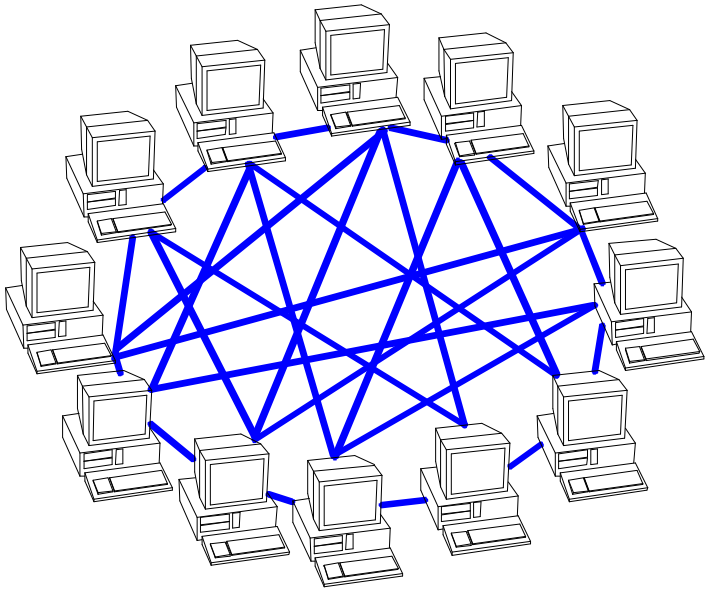


Centralised P2P System by Example: Napster

- Early file/music sharing system
- Central file-to-peer index server operated by Napster
  - Peers lookup other peers providing certain MP3 files by querying server
  - Pure client/server communication between peers and server
- File transfer: directly between peers via HTTP
  - Shared resource: files
  - P2P communication to access distributed files

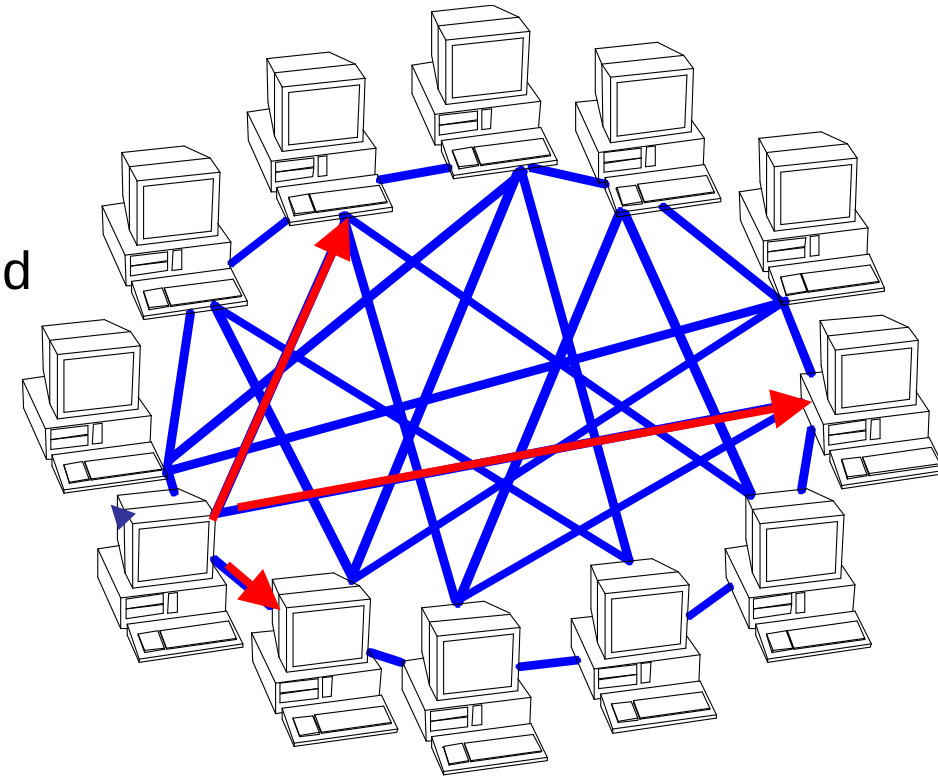


- Fully distributed functionality among peers
  - Distributed resources and direct P2P resource access
  - Fully distributed system functionality
    - Distributed peer lookup and routing
  - No single point of failure, *any* component can be removed without loss of functionality



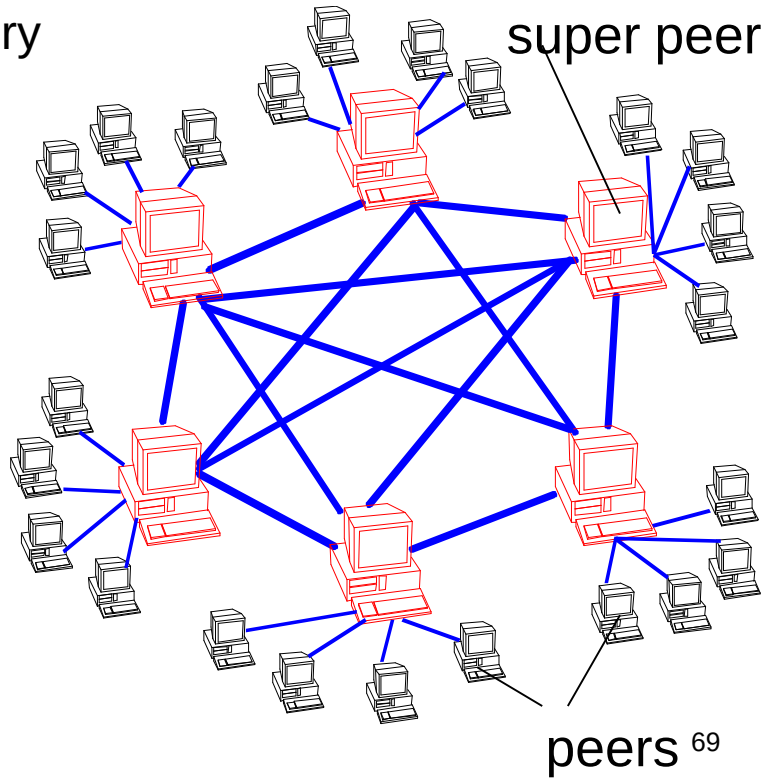
Pure Peer-to-Peer System: Gnutella v0.4

- Distributed file sharing system
- Overlay network between peers
  - Each peer connected to limited number of peers via TCP
- Searching: limited flooding
  - Peers forward new queries to neighbours until content is found
  - Restricted path length/time to live of query
  - Response send via reverse path



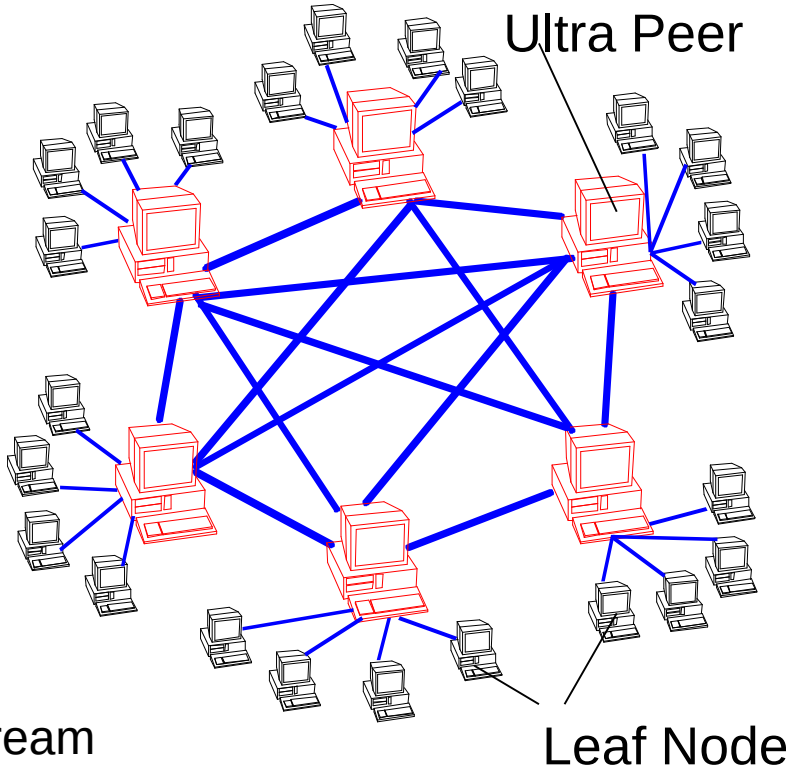
Hierarchical “Super” Peer-to-Peer Systems

- Two classes of peers:
  - (Ordinary) peers: provide shared resource, e.g. content
  - Super peers: lookup of peers
    - Criteria to become super peers, e.g.: processing power, bandwidth, memory
    - Set of super peers is dynamic
- 2-level hierarchical network
  - *Level 1: peer/super peer network*
    Each peer connects to one super peer, 1 super peer serves *n* peers
  - *Level 2: super peer network*
- Hybrid of centralised P2P system and pure P2P system
  - Level 1: centralised P2P system
  - Level 2: pure P2P system



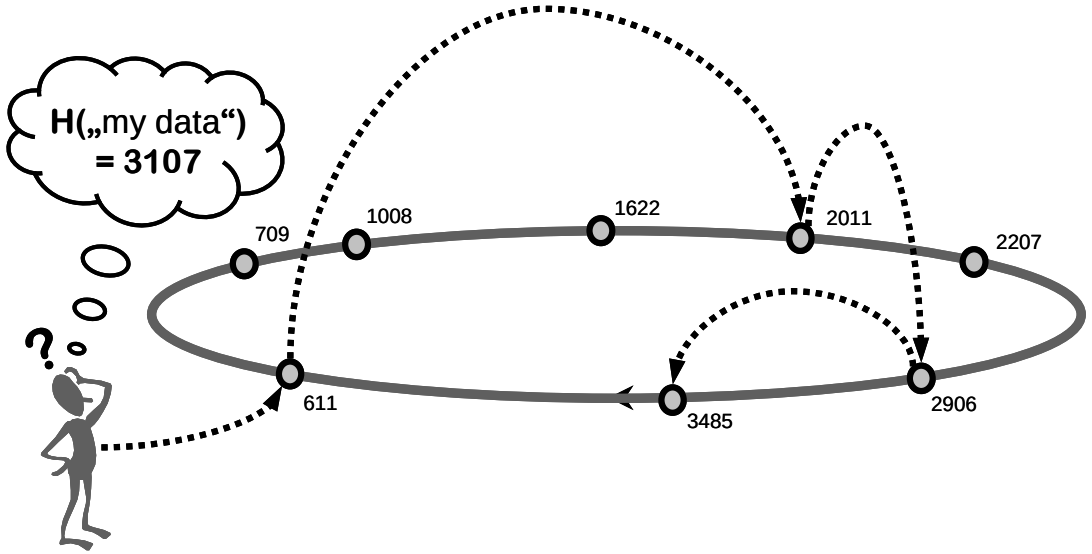
Super P2P System by Example: Gnutella 0.6 (1)

- Two classes of Gnutella nodes:
  - Leaf Nodes (= peers)
  - Ultra Peers (= super peers)
- Ultra Peer selection criteria:
  - Not firewalled
    - Incoming connections possible?
  - Suitable operating system
    - Large numbers of sockets required
  - Sufficient bandwidth
    - > 15KB/s upstream, 10 KB/s downstream
  - Sufficient uptime
    - At least a few hours
  - Sufficient RAM
  - Sufficient CPU speed



- Network topology
  - Leaf Node connected to single (or few) ultra peers
  - Ultra Peer connected to ...
    - ... large number of leaf nodes (10-100)
    - ... small number of Ultra Peers (<= 32)
- Query Forwarding
  - Leaf node sends summary of its content to its Ultra Peer
    - Hashed resource names
  - Ultra Peers builds local routing table from summary
    - Routing table is not forwarded to other Ultra Peers
  - Forwarding between Ultra Peer and Leaf Node:
    - Incoming queries are matched against routing table
    - Queries are forwarded to nodes that may provide content
  - Forwarding between Ultra Peers:
    - Limited flooding (see Gnutella 0.4)

- Content placed at specific locations (peers)
- Defined relation between content of peer and location in overlay network
  - Each peer is responsible for a defined subset of content
- Overlay network is structured according to peer content
  - Directed routing towards peer providing content (or link to content)



Further topologies:

- Multi-level (#levels > 2) hierarchical networks
- Further hybrid topologies
  - The term *hybrid* is overloaded!
  - Already presented some hybrid Peer-to-Peer systems
    - Centralised Peer-to-Peer system:
      → Hybrid of client/server and P2P system
    - Super Peer-to-Peer system
      → Hybrid of centralised and pure Peer-to-Peer system
  - Combination of structured & unstructured networks
  - And many more…