UniS

CS254 Network Technologies

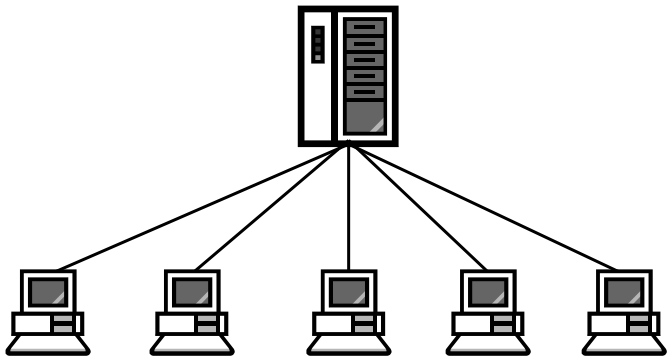
Lecture 9: Peer-to-Peer Networks I

James Salter
Department of Computing
University of Surrey
Autumn 2006

What is "P2P Computing"?

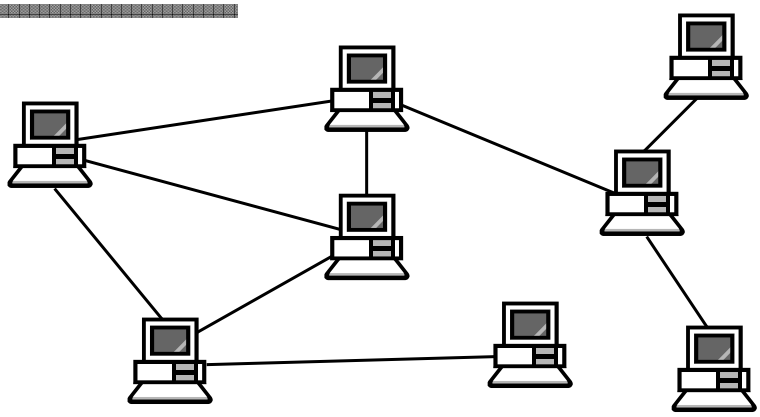
- Everybody seems to have a different idea
  - File sharing
  - Distributed computing
  - Inter-connected communities of users
  - BotNets
  - Opposite of client/server network
- These are all valid definitions

Client/Server Architecture



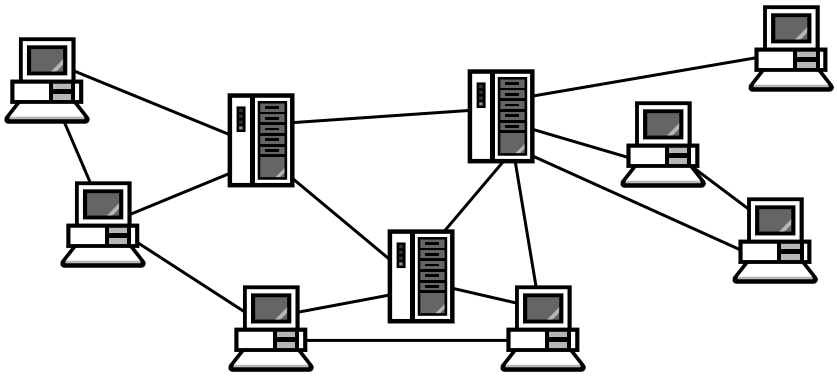
- Clients send requests to/via a central server
- Small #messages
- Single point of failure

Pure Peer-to-Peer Architecture



- No central server
- Clients connected to one or more peers
- Resilient
- Large #messages

Hybrid Peer-to-Peer Networks



- Some nodes act as group managers/routers
- More robust than client/server
- Potentially lower #messages than pure P2P

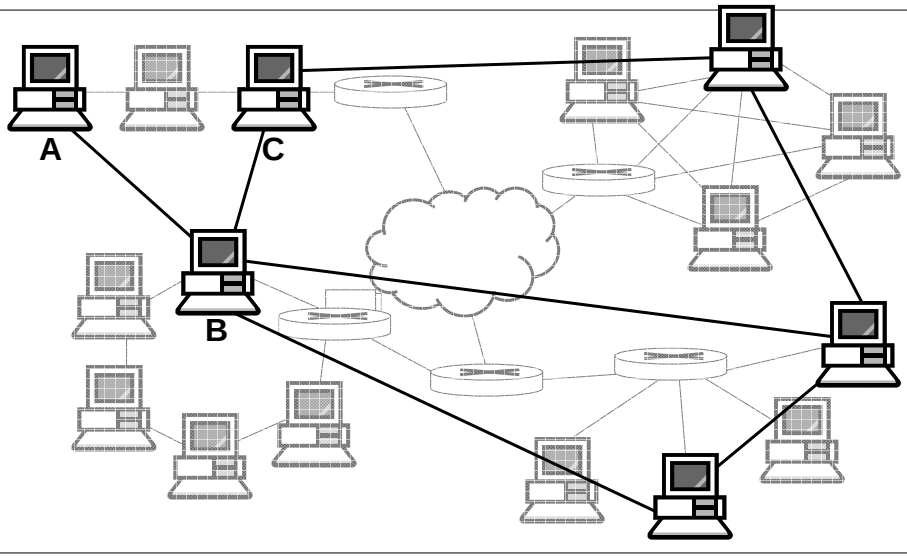
Characteristics of a P2P Network

- Nodes can act as clients and servers
- No centralised server/authority
  - (in pure P2P networks)
- Network is highly dynamic
  - Nodes join and leave regularly
- Large-scale
  - Potentially millions of nodes
- Nodes are autonomous
  - But co-operate to share/retrieve resources

Overlays

- Creates a virtual network: network overlay
- Connections between nodes defined by application, not by physical connections
- Primarily concerned with nodes: ignore routers, bridges, other networking devices
- Overlay can have little relation to underlying network connections
  - Needlessly send data across the Internet instead of through a LAN?

Overlays

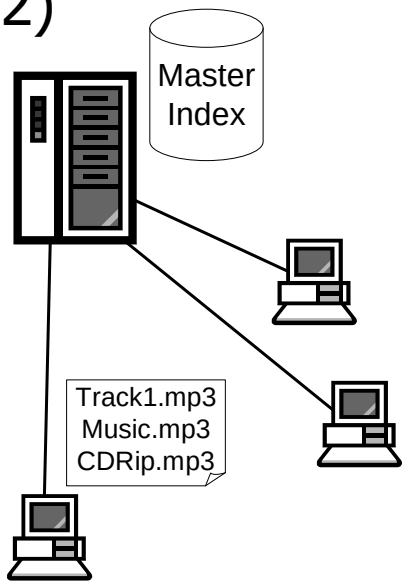


Resource Discovery Problem

- Files/resources stored at the "edge" of the network
- No single location hosts resources
- Resources can be shared by anybody
- How do we know whether the resource we want exists?
- How do we know where to find the resource we want?

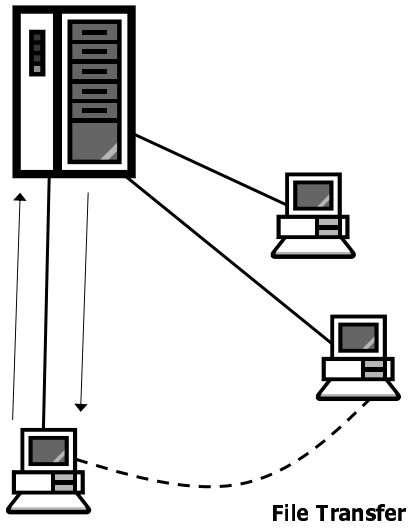
Napster (1999-2002)

- All nodes connected to the Napster server
- Nodes send a list of files they are willing to share to server
- Server hosts a master index of files available on the network, and which machines hold what



Napster: Downloading a File

- Send query to Napster server
- Server returns addresses of machines hosting file
- Set up a direct connection with machine to download: file does not pass through Napster server

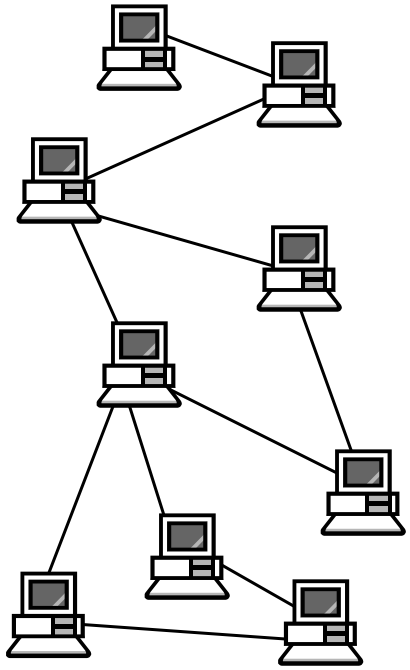


Napster: Advantages/Problems

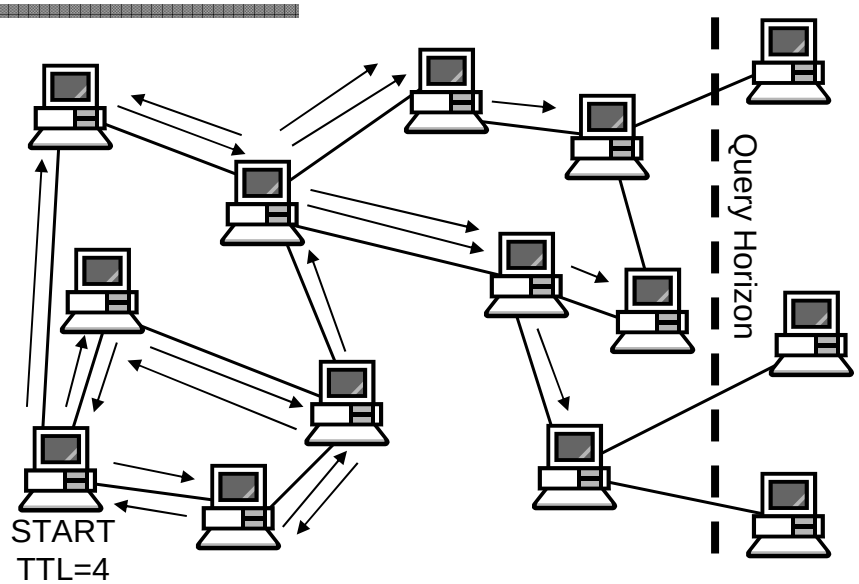
- Small number of messages to find a file
- Guaranteed to find file if it exists
- Index server is a single point of failure:
  - Bottleneck – potential limit to scalability
  - Attacks – attract Denial of Service attacks, etc.
  - Control – single entity can decide to shut down
  - Legal – all query traffic passes through single machine: owners cannot deny knowledge of what network is being used for

Gnutella

- Nodes directly connected
- No central server
- Query flooded from node to node
- Time-To-Live (TTL) counter prevents infinite query forwarding:
  - Decremented at each node
  - Query discarded when TTL=0



Gnutella: Query Forwarding

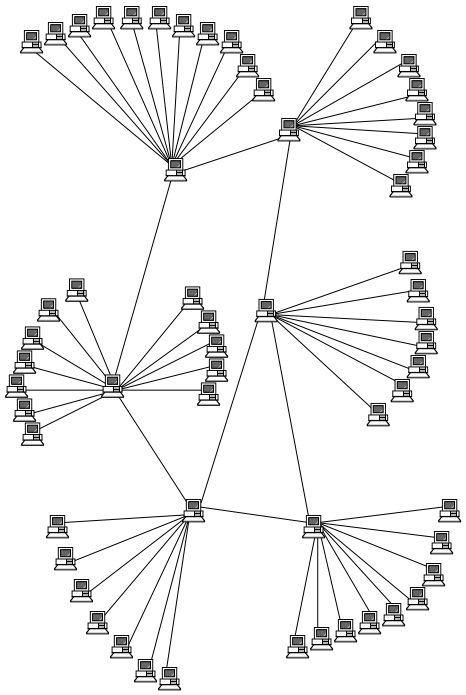


Gnutella: Advantages/Problems

- No single point of failure or central controller
- Difficult/impossible to shut down
- Large volumes of query traffic: query must be sent to every node individually
- Wasted messages: some nodes receive query multiple times
- Query horizon (TTL limit): cannot guarantee to find file in network

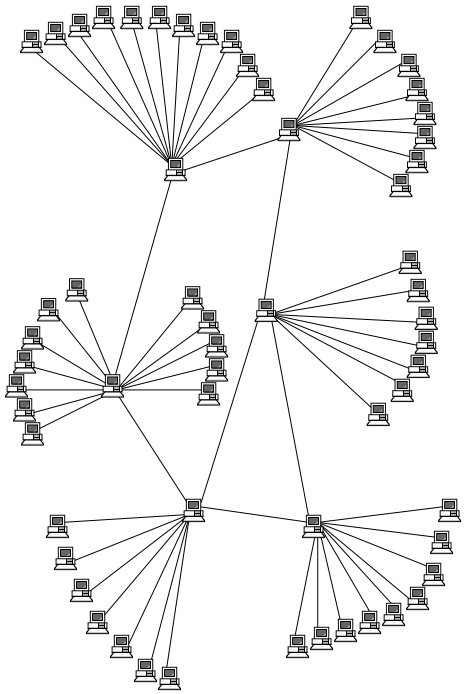
Gnutella2

- Leaves: standard nodes connected to 1 or 2 hubs
- Hubs: super nodes connected to 100s of leaves and several other hubs
- Hubs hold index of files held on each leaf node connected to them
- Hubs share indexes between each other



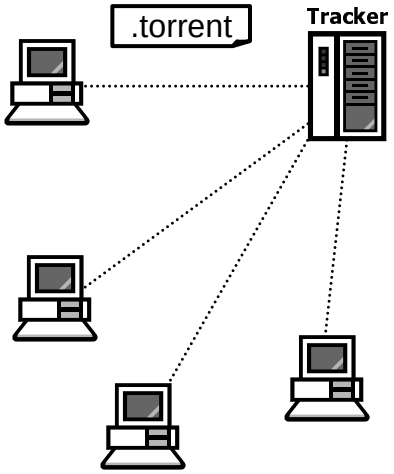
Gnutella2: Search

- Contact hub: query forwarded to nodes listed in index as hosting file
- Hub forwards query to other hubs where file is found in their index
- Can find popular file without loading the network
- More likely to find rare files?



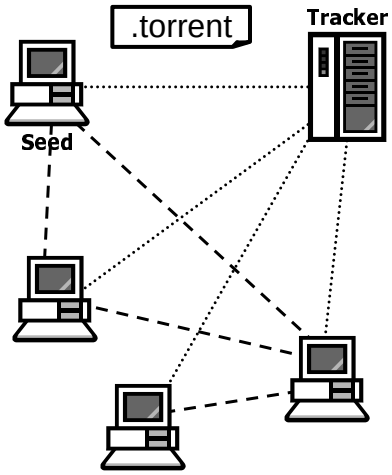
BitTorrent

- No search mechanism
- No permanent network
- Must know location of .torrent file
- .torrent contains file info plus URL of a tracker
- Tracker sends list of machines currently downloading file



BitTorrent

- File is split into pieces (typically ¼Mb)
- Download different pieces from each peer
- Share the pieces already downloaded
- Several techniques for prioritising order
- Must have "seed" machine with complete copy of file



Structured P2P

- Avoid random visiting of nodes to find content
- Utilise an index similar to Napster
- Split index and distribute over entire network
- Must know where to find each piece of the index – organisation of nodes
- More efficient (less hops): requests always routed *towards* content

Hash Tables

- Hash functions map data keys to buckets
- Keys are stored in buckets in index table
  - Example:

f(*x*) = *x* mod 6

f(15) = 15 mod 6 = 3
f(36) = 36 mod 6 = 0

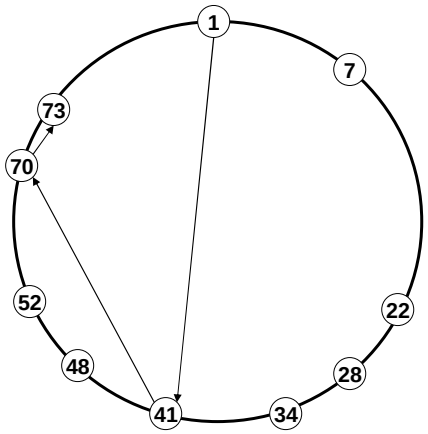
0	12, 24, 30
1	19, 61
2	20, 26, 56
3	3, 9
4	10, 16, 28, 34
5	23, 35, 65

Distributed Hash Tables

- Each node hosts part of the index (one or more buckets)

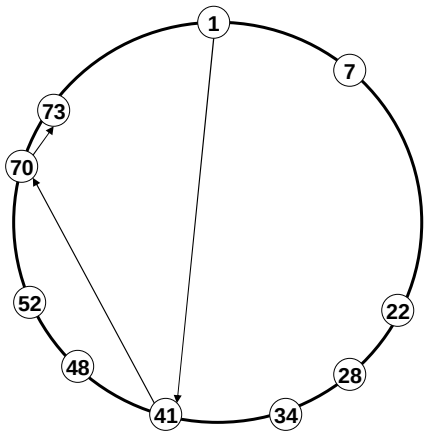
Chord

- Each node has a unique ID number
- Nodes arranged numerically in a ring
- File names hashed to generate file ID
- Files hosted on node with closest ID
- Lookup: provide key, returns value



Chord: Searching

- Nodes host small routing tables
- Each entry in table points further away from current node
- Allows query to be forwarded quickly through ring
- Distance between current node and target halved each step
- Details next week…



Comparison with Unstructured Nets

- Guaranteed to find resource if it exists
  - No TTL counters/Query Horizon
- No wasted messages: lookup sent towards answer
  - Query may visit same node multiple times in Gnutella
- More scalable lookup costs
  - DHTs typically O(log *N*) vs Gnutella O(*N*)
- Can only lookup value associated with key
  - No support for complex queries (e.g. search within file)
- Must regularly maintain routing information
  - No/little maintenance necessary in Gnutella

P2P Applications

- File sharing
- Distributed computing
- Instant messaging
- Media streaming/distribution

- Web services
- Workgroups
- Networked devices

Research Areas

- Resource discovery/searching
  - Efficiency, more powerful queries
- Robustness/stability
  - Can the network operate under high churn?
- Incentives
  - Why should I provide as well as consume?
- Reputation/Quality
  - How good/reliable are peers in the network?
- Security
  - Malicious/suicidal peers, attacks on the network

Further Reading

- Oram, A., *Peer-to-Peer: Harnessing the Power of Disruptive Technologies,* "Chapter 1: A Network of Peers". Free download from: http://www.oreilly.com/catalog/peertopeer/chapter/ch01.html
- http://en.wikipedia.org/ for peer-to-peer, Gnutella (original and subsequent versions), Napster, etc.