Peer-to-Peer Networking

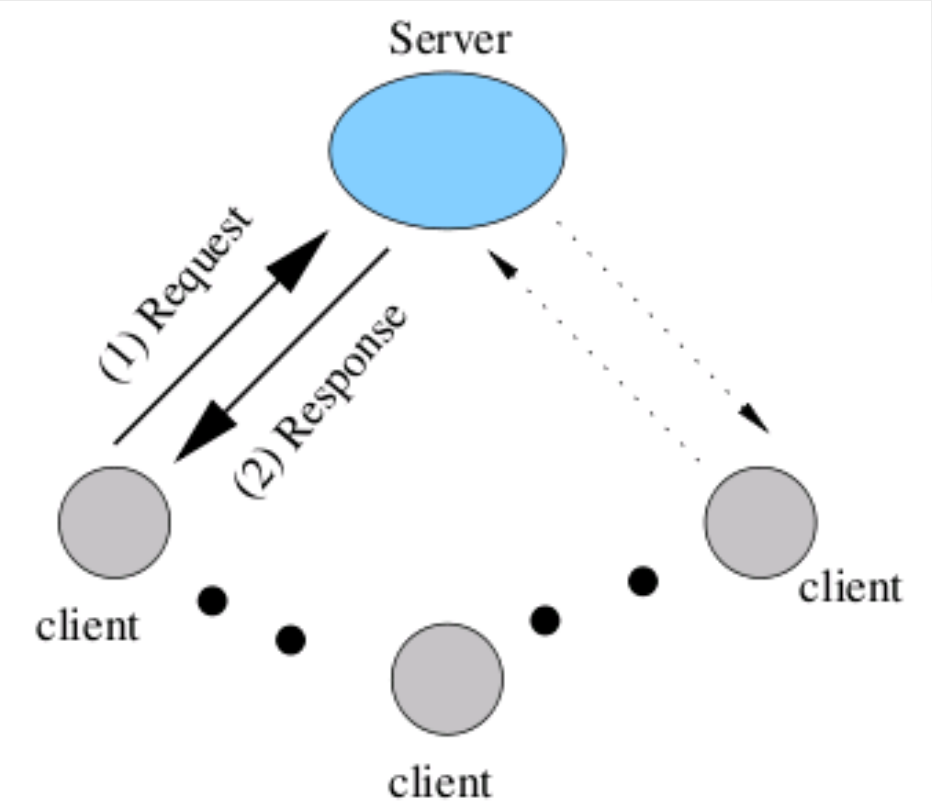
P2P Applications and Systems

Lecture Content

- Introduction to the most common P2P systems
- 4 areas according to application environment perspective
  - Identifying basic characteristics of P2P applications
- Closer study of P2P application classes
  - Example applications examined from each class
- But first, let's recall the centralised client-server system...

Centralised Systems

- Recall the old server-client model
- Easy and clear management
- Centralised means that all communication and computing is done in a single dedicated server
- Limited scalability: adding multiple servers cannot solve the linear growth of network bandwidth
- Scaling computing expensive
- Single point of failure
- Possible to scale effectively with lots of money (Google)



[Kant et al., A Framework for Classifying Peer-to-Peer Technologies]

P2P Applications and Systems

- P2P Content sharing
  - Typically file sharing: Napster, Gnutella, Kazaa
- P2P Hardware resource sharing / computation
  - Computing cycles, SETI@home
- P2P Communication
  - Instant messaging and Skype
- P2P Collaboration
  - Enabling peers to work jointly, online encyclopaedias

P2P Content Sharing

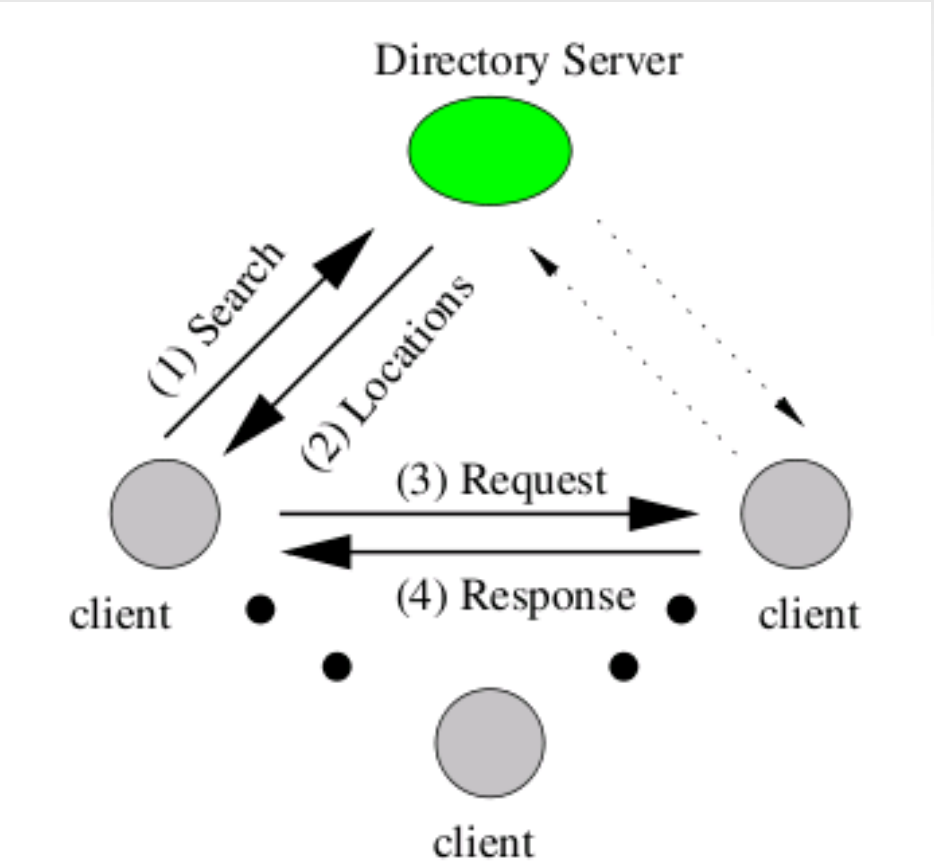
- File sharing most popular type: Napster, Gnutella, ...
  - Searching and downloading individual files from peers
  - Digital media (music, video) and other file types
  - Napster: first killer application which made P2P popular
- P2P content publishing and storage systems (Freenet)
- Broad range of applications
  - Relatively simple direct file sharing applications
  - Sophisticated systems: providing security, anonymity, efficient search and distribution mechanisms

Napster

- The first P2P file sharing application (1999-2002)
  - Though provided only distribution for MP3 files
  - 26 million users worldwide in February 2001
- Original Napster closed eventually in 2002
  - Most of the content were under copyright protections
  - Common challenge to many other P2P file sharing
  - Nowadays Napster is a legal online music store

Centralised P2P System

- Napster used centralised directory service (can be multiple servers)
- Maintains peer's address, shared files and metadata
- Request sent to server, which looks through the index
- Reply contains information which peers have matching files
- Download directly from a peer
- Napster is a *hybrid* P2P system: a subset of services are centralised while peers communicate directly



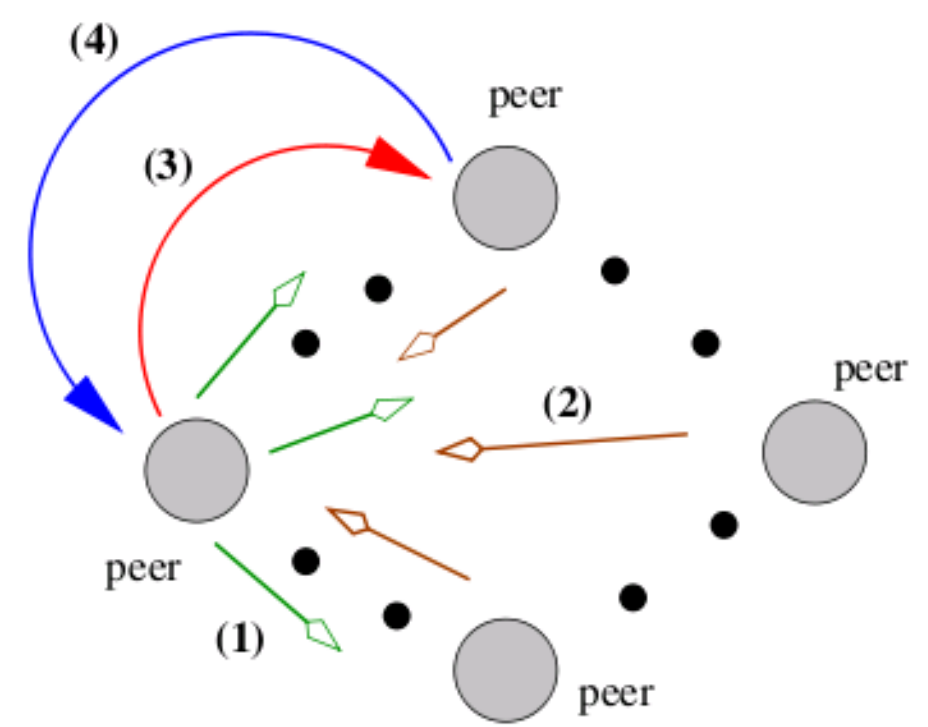
[Kant et al., A Framework for Classifying Peer-to-Peer Technologies]

Napster Features

- Centralised service
  - Searching efficient and precise: has all files listed and knows where to get them
  - Computing (search) unscalable (e.g. to millions of peers)
  - Single point of failure: server failure prevents service
- Content distribution (file sharing) moved to the edge of the network: balances loads on the network
- No guarantees for user provided data integrity
  - Common problem to most of P2P applications

Decentralised P2P Systems

- Searching a data item: query neighbours within a certain radius
- When successful reply received, request a transfer from that peer
- Peer responses with the data item
- Extremely resilient to random node failures and peers entering and leaving the system
- Challenge: how to find items efficiently (i.e. route queries), when only little information about network is known, some neighbours



(1) Search, (2) Location
(3) Request, (4) Response

Gnutella

- Tried to avoid weaknesses of centralised P2P
- Idea: fully distributed, ad-hoc network of peers
- The initial version of Gnutella was pure P2P
  - peers called servents, acting both as servers and clients
- Later versions introduced a hierarchical structure
  - Upper layer of super-peers dedicated to searching

Gnutella: Search

- Search implemented as flooding query
  - Send query to all neighbours
  - Neighbours forward query until set TTL value reached
  - Reply back-propagated on reverse path of broadcast
  - Prevent loops by remembering recent broadcasts
  - Not scalable, causes congestion on network
  - Saturated connections drop queries
  - Too short TTL cannot reach far (i.e. item not found)

Gnutella: Protocol

- Get a Gnutella peer IP address (out-of-band method)
- Handshake with GNUTELLA CONNECT, replied with OK message
- “PING”: announce your presence to other peers, neighbours flood PING forward
- “PONG”: back-propagated reply with information about their addresses and their shared data items
- PING sent periodically to update network

Gnutella: Protocol

- Search with QUERY and user specified string
- Compare received QUERY string to locally stored files and reply with QUERY RESPONSE and information necessary to download a file
- GET: file transfer directly between peers (HTTP)
- PUSH request can be sent to avoid firewall blocking
  - Peer inside firewall has open TCP connections to outside
  - PUSH can be routed inside, file sent directly to requester

Gnutella: Hierarchical Version

- Proposed version 0.6, advances to query technique
- Ultra-peer (super-peer) manage querying of a subnet
- Joining peer sends file list to super-peers
- Peer sends its query to a super-peer
- Ultra-peer floods query to neighbour super-peers
- Dynamic query: flood again deeper, when no reply

Gnutella: Discussion

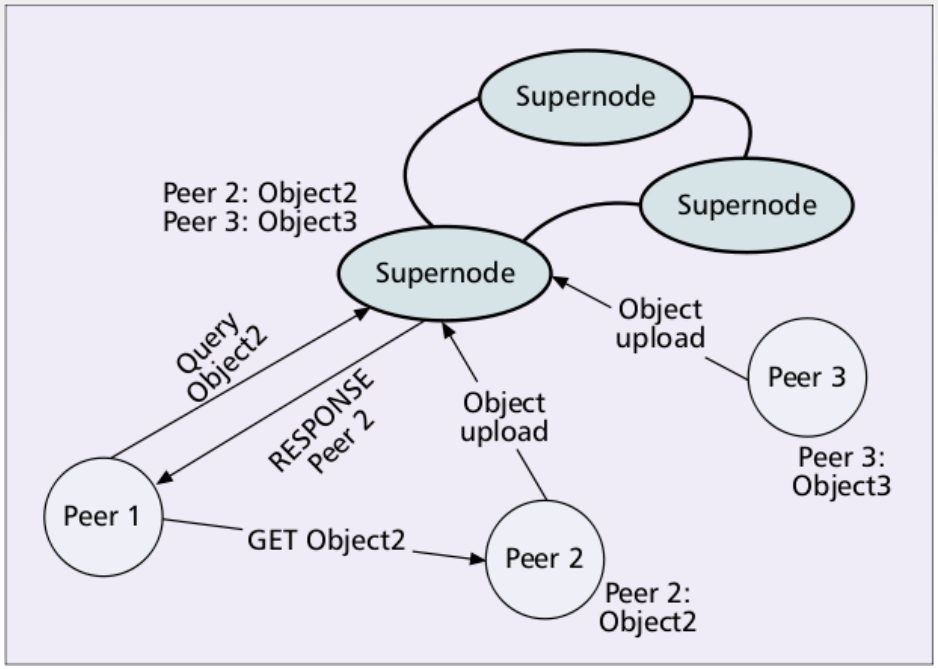
- Original version 0.4 flooding query inefficient
  - Network congestion
  - Cannot guarantee hit, search limited by TTL
  - Periodic flooding PING wastes resources
- Newer version 0.6 still a proposal, but in practise the most commonly used
  - Hierarchical structure with ultra-peers scales better
- Suggested a cluster-based architecture where delegate peers act as an directory servers

FastTrack/Kazaa

- Kazaa is yet another file-sharing system, based on proprietary FastTrack protocol
- 2nd generation P2P system, similar to Gnutella 0.6
  - Two-level hierarchy: super-peers on top of normal peers
  - Super-peers having high bandwidth, CPU, memory, etc.
- Concept of super-nodes sometimes called as partially centralised P2P system, or distributed indexing

FastTrack/Kazaa Routing

- Designated super-peers (supernodes) with greater resources
- Normal peers upload their files meta-data to super-peers
- Query to super-peer, download directly from the peer according to the reply
- Super-peers perform flooding queries in a highly pruned super-peer overlay network in order to locate the searched object



[Lua et al. A survey and comparison of peer-to-peer overlay network schemes]

FastTrack/Kazaa Discussion

- Introduced a couple of important P2P issues
  - P2P systems are naturally very heterogeneous containing peers with different resources: set “powerful” peers to do work which benefits all
  - Organised hierarchy results in more efficient searching
- Search still based on flooding between super-peers
- Popularity: took a major share of file sharing users
- Known to contain malware and spyware
  - The license terms require user to install 3rd party software

P2P Computation

- Computation or “cycle sharing” is the most common P2P hardware resource sharing type
- A typical desktop PC mostly idle and could be harnessed for solving complex computational tasks
- Decomposing the problem into large chunks which can be run with very little interaction (sometimes referred as “embarrassingly parallel” computing)
- P2P hardware resource sharing include, e.g., (replicated) distributed data storage

SETI@home

- Probably the best known cycle sharing application
- Search for extraterrestrial intelligence by analysing radio signals from space
- Simple computing model: the signal data is divided into fixed-size work units distributed via the Internet to a client program running on numerous computers
- The client program computes a result (a set of candidate signals), returns it to the server, then gets another work unit (no messages between clients)

SETI@home

- Signal data analysis depends on computing power
  - Large bandwidth, radio noise from Earth, Doppler effect
  - Recognise artificial signals? Pulses, spikes, Gaussian rises
- Redundant computation
  - Each work unit is processed multiple times
  - Detect faults from hardware and malicious users
- 5 million participants worldwide, the largest computation in history, over 2 million years of aggregate computing time

SETI@home Implications

- Public-resource computing
  - Shifting resource-intensive functions from central servers to workstations and home PCs
  - Computing time? SETI@home work unit 10 hours or so
  - Minimise network traffic (receive 350KB, send 1KB)
  - Nodes cannot be trusted (redundant computing)
- Heterogeneity of peer: wide support of platforms
  - Develop general-purpose frameworks for public-resource and other large-scale distributed computing

Is SETI@home Really P2P?

- The use of central control makes the SETI@home project in conflict with the idea of pure P2P computing
- Still this is recognized as peer-to-peer, because it supports the idea of contributing autonomous peers working together as one system
- Most distributed computing projects (all the big ones) are today based on centralised coordination

Non-Technical Details

- A public-resource computing project must attract participants... What made SETI@home so popular?
  - Users do not benefit from giving away their (idle) cycles
  - Common goals seem to appeal people
  - “Find ET for the good of humanity”
- Non-profit, scientific research
  - Would commercial/production/military use be interesting?
- Relies primarily on mass-media news coverage and word-of-mouth to attract participants

P2P Communication

- Equality: enables participants to initiate, manage and terminate the session
- Instant messaging (IM) very popular
  - Short messages directly between peers, or “buddies”
  - Centralised server holds user account data
  - Server can tell which of your peers are online and their addresses, may also help to initiate transfers over firewall
  - Can transfer binary data and audio/video stream
  - Examples: AOL, ICQ, MSN, etc.

Skype

- Skype is a good example how P2P can be applied to basic services, such as telephony
- Offers Voice-Over-Internet Protocol (VoIP) and IM
  - Call other user on Internet
  - Call regular telephone numbers (VoIP-PSTN gateways)
  - Receive calls from regular telephone network
- Popular, user-base of 50 million users
- Proprietary network and encrypted communication

Skype: Known Implementation

- Not much really known about Skype internals
  - Similarities to Kazaa (and same developers)
- Super-node based hierarchical P2P network
  - Super-nodes elected by some criteria (resources?)
- Normal nodes send control traffic, queries and IM over the super-node P2P network
  - Super-node can help routing through firewall and NAT
- VoIP and file transfer session are peer-to-peer

Skype Discussion

- User lacks some of the controllability
  - Acting as super-node not easily prevented
  - User may not want to give all bandwidth to Skype
- Can travel through many firewalls and NATs
- Proprietary software concerns
  - Prevents developing compatible clients
  - Possible security holes/back-doors? Cannot say for sure
- Succeeded in P2P real-time communication

P2P Collaboration

- P2P collaboration happens between people
- Applications may include sharing of content and joint computations
  - Creation of online communities
  - Development projects
  - Computing communities
  - Gaming communities
  - Joint virus protection

Examples: P2P Collaboration and Communities

- Marketplace
  - Integrating sellers, suppliers and vendors with their existing tools and business models into a P2P framework
  - Products and provider information kept on the peers
- Interest-based community (OpenCola Folders)
  - Conducting effective searches on the Internet
  - Contextual searching and similarity - “John's document helpful, probably other John's documents are interesting. And John has related references to Cindy's documents”

P2P Collaboration in Online Development Projects

- Example of application integration development
- Ford Motor Company needed efficient collaboration to build fuel-efficient auto mobiles
  - Multiple functional groups involved present different vehicle attributes: weight, transmission, electrical, etc.
  - Fuel economy group receives input data from the groups, aggregates it and runs numerous trade-off scenarios
  - Data remains local to its source, but new tools enabled easy access: engineers communicate directly (P2P)
  - Total time dropped from 2-3 days to 23 seconds

Summary

- P2P can be applied to numerous problems
- File sharing (content distribution) most popular
- Solutions differ according to objectives
  - File sharing: (de)centralised, distributed, hierarchical with super-nodes
- SETI@home and other similar computing projects
- Collaboration & communication also in many flavours

Summary

- They all present some subset of the P2P principles
  - Sharing resources, such as content and storage
  - Sharing computational power
  - Computation moves to the edges of the network
  - Network nodes, peers, are more autonomous
  - Coping with dynamic nature and heterogeneity of wide networks

Consider the Following

- How have file sharing applications (architectures) developed over the last years?
- Is there any solution raising above others, or just a bunch of more or less working applications?
- Would you list *Wikipedia*, the online encyclopaedia, as some kind of P2P system?
  - Why? Why not?
  - What type of P2P would it be?