# Distributed Software Development

## Peer-to-Peer

Chris Brooks

Department of Computer Science
University of San Francisco



## Peer-to-Peer

- Peer-to-peer is often touted as a hot new technology
  - Really, it's as old as the Internet (or earlier)
- Fundamental idea: all nodes in a network should be capable of the same functionality.
- Contrast this with client-server.
- In practice, many systems are a hybrid of p2p and client-server.



## Examples

- What are some examples of p2p systems?



## Examples

- File sharing apps (Morpheus, Kazaa, Limewire, etc)
- IRC/ICQ
- SETI@Home/distributed.net
- Web caching
- VoIP
- SMTP
- etc



## Historical examples

- USENET
  - Peers within a domain would collect usenet posts and forward them via UUCP, then NNTP.
  - No central server or repository.
- DNS
  - A hierarchical p2p system



## client/server

- Why did the client/server model become so popular?
  - Asynchronous network structure
  - Ease of discovery
  - Centralized content development
  - Firewalls, NAT make it harder for users to host content.
  - Model changes: fewer developers, more browsers.

## Potential advantages of peer-to-peer

- Reduce server bottlenecks
- Move content closer to users
- Allow users to publish as well as consume information
  - Note that publishing and authoring can be different things.
- Scalability
- More able to deal with adaptive, dynamic networks.



## Challenges of peer-to-peer

- Finding peers (addressing)
  - How does a peer join the network?
  - How do peers identify each other?
  - Is the network constructed statically, or dynamically?
  - Does a peer always connect to the same set of peers?



## Challenges of Peer-to-Peer

- Searching for data
  - Naming conventions
  - How are queries distributed?
  - Are all peers searched, or just a subset?
  - Do peers keep track of the contents of other peers?



## Challenges of Peer-to-Peer

- Other challenges:
  - Interoperability and standards
    - Currently, every P2P network has its own protocol – you can't use a BitTorrent client to search on Gnutella.
  - Dealing with dynamic networks
  - Security
    - This includes the distribution of malicious files, protection from snooping, and user authentication.



## First-generation applications

- So-called first-generation applications use a centralized name server.
- All user addresses and file indexes are kept there.
  - This can be used to build an application-level address-resolution protocol.
- Napster is the canonical example of this.
- Most IM programs also work this way.
- Central server is used to search.
- Files are transferred between peers.



## First-generation weaknesses

- Central server provides a bottleneck.
- Difficult to implement in a closed or dynamic environment.
- Scalability can be a problem.
- (also, legal issues in the case of Napster)

## Second-generation models

- Second-generation models remove the central server.
- All file information is distributed.
- Advantage:
  - Potentially more scalable
  - Legal issues avoided.
- Disadvantages:
  - Search is more complex
  - Discovery of peers more complicated.



## Example: Gnutella

- Gnutella is an example of an open P2P system.
- Many clients use the same protocol
  - LimeWire, BearShare, Gnucleus, etc
- Gnutella uses no centralized server - all information is stored at the peers.



## Example: Gnutella

- The Gnutella network is completely decentralized.
- When a peer comes onto the network, it sends a ping to all known peers.
- These peers respond with a pong, which identifies all their known peers.
- Your peer can also remember previous connections.



## Example: Gnutella

- When a peer wants to perform a search, it sends a query to all known nodes.
  - This request is then forwarded on to nodes one level away, who forard it to all nodes they know about, and so on.
  - This is known as query flooding.
  - If a hit is found, the node containing the file contacts the searcher and download begins.



## Example: Gnutella

- Problems with Gnutella:
  - Searching is unreliable - the network is often partitioned.
  - Search based only on keywords - this produces naming issues.
    - If I want "Lost S1, Ep 3", what do I search for?
  - Wastes bandwidth forwarding searches to all nodes.
  - Does not take advantage of network structure
    - Some peers may have better bandwidth, more information, or a more central network structure.



## Hierarchical p2p

- Kazaa is an example of a hierarchical p2p system
- When a user enters the network, it connects to a meganode.
- This meganode forwards queries to all other nodes connected to it.
- It also collects file information for each client.
- Periodically, it also exchanges information with other meganodes.
- Again, searches typically do not reach the entire network.
- Higher-bandwidth nodes act as meganodes, process more traffic.