

CS162
Operating Systems and
Systems Programming
Lecture 27

P2P Networks

May 12, 2008
Prof. Anthony D. Joseph
<http://inst.eecs.berkeley.edu/~cs162>

Goals for Today

- P2P Networks

Note: Some slides and/or pictures in the following are adapted from slides by J. Pang, B. Richardson, I. Stoica

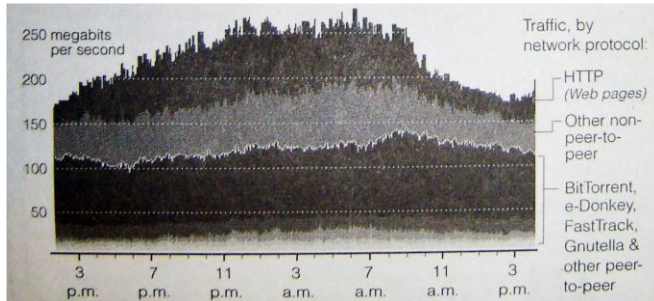
5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.2

Why Study P2P?

- Huge fraction of traffic on networks (>=50% in '05)



- Exciting new applications
 - File sharing, distributed computing, Voice-over-IP, Video
- Low cost access to 10's-100's of TB of data

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.3

What is P2P used for?

- Use resources of end-hosts to accomplish a shared task

- Typically sharing files
- "Telephone" calls
- Watching videos
- Playing games
- Searching for patterns in data
 - » Seti@Home, Folding@Home, ...



- Take advantage of resources at the edge of the network

- Fundamental shift in computing/networking capabilities
- Sum of distributed computing » central computing
- Increase in total bandwidth versus central servers

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.4

Peer to Peer Systems

- **Napster**
 - First music file sharing service (June 99-July 01)
- **Gnutella**
 - First open-source music file sharing service (early 00-)
- **Kazaa/Skype/Joost**
 - KaZaA (easy file sharing - 2001)
 - First Skype beta (easy VoIP - Aug 03)
 - Joost beta (easy online video - 2006)
- **BitTorrent**
 - Designed by Bram Cohen in April 01 (1st SW July 01)
 - 18-35% of all Internet traffic (2004)

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.5

Key issues for P2P systems

- **Join/leave**
 - How do nodes join/leave? Who is allowed?
- **Search and retrieval**
 - How to find content?
 - How are metadata indexes built, stored, distributed?
- **Content Distribution**
 - Where is content stored? How is it downloaded and retrieved?
- **Four Key Primitives**
 - Join: How to enter/leave the P2P system?
 - Publish: How to advertise a file?
 - Search: How to find a file?
 - Fetch: How to download a file?

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.6

Publish and Search

- **Basic strategies:**
 - Centralized (Napster, BitTorrent)
 - Flood the query (Gnutella)
 - Smart/partial flood the query (Kazaa/Skype/Joost)
- **Different tradeoffs depending on application**
 - Robustness, scalability, legal issues

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.7

Napster: Overview

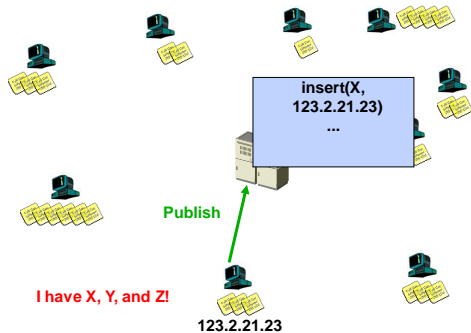
- **Napster (June 99-July 01)**
 - Created by Shawn Fanning (Northeastern undergrad)
 - Peak Feb 01 (26.4M total/1.5M simultaneous users)
 - Shutdown due to lawsuits, now reborn
- **Centralized Database:**
 - Join: on startup, client contacts central server
 - Publish: reports list of files to central server
 - Search: query the server → return someone that stores the requested file
 - Fetch: get the file directly from peer

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.8

Napster: Publish

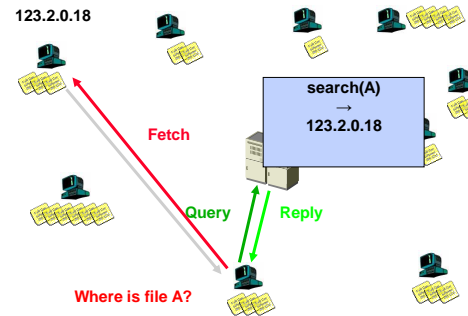


5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.9

Napster: Search



5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.10

Napster: Discussion

- **Pros:**
 - Simple
 - Search scope is $O(1)$
 - Controllable (pro or con?)
- **Cons:**
 - Server maintains $O(N)$ State
 - Server does all processing
 - Single point of failure

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.11

Administrivia

- No office hours on Tue 5/13
- Project #4 code deadline is Wed 5/14 at 11:59pm
 - Don't store files in `/home/<user>`, use `/tmp` instead
 - Use `ssh` to diagnose login problems
- Final Exam
 - May 21st, 12:30-3:30pm
 - Exam will be comprehensive
 - » All lectures, papers, projects, class/section discussions
 - Closed book, notes, slides
 - One 2-sided cheat sheet allowed

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.12

Gnutella: Overview

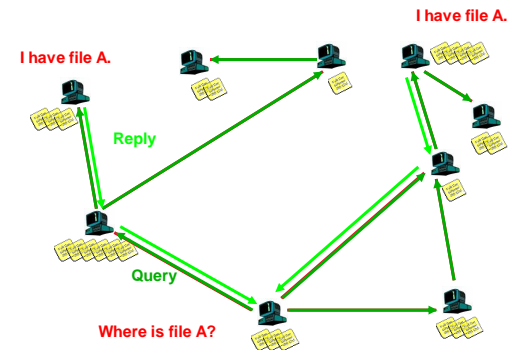
- Released in 2000 by J. Frankel and T. Pepper from Nullsoft
- Soon many other clients: Bearshare, Morpheus, LimeWire, etc.
- In 2001, many protocol enhancements including "ultrapeers" (more on this later...)
- Query Flooding:
 - Join: on startup, client contacts a few other nodes; these become its "neighbors"
 - Publish: no need
 - Search: ask neighbors, who ask their neighbors, and so on... when/if found, reply to sender
 - Fetch: get the file directly from peer

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.13

Gnutella: Search



5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.14

Gnutella: Discussion

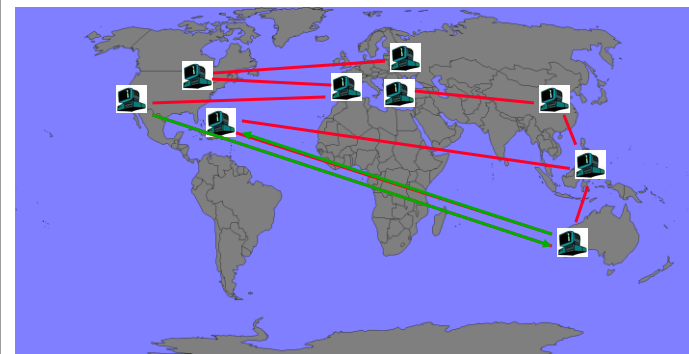
- Pros:
 - Fully de-centralized
 - Search cost distributed
- Cons:
 - Search scope is $O(N)$
 - Search time is $O(???)$
 - Nodes leave often, network unstable

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.15

Aside: Search Time?

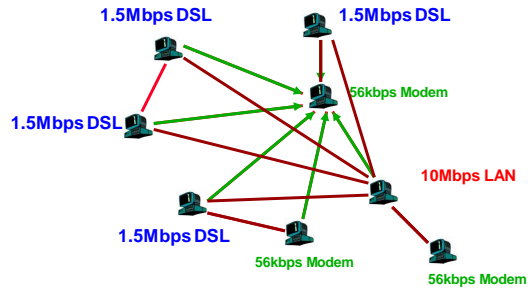


5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.16

Aside: Are All Peers Equal?



5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.17

Peers are Different

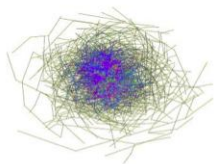
- Available Bandwidth
 - FastTrack: 33% PCs have mean downstream b/w 56Kbps or less, 50% have mean upstream b/w \leq 56Kbps
 - Direct Connect: 20% PCs have mean downstream b/w 56Kbps or less, 33% have mean upstream b/w \leq 56Kbps
 - » (Sen and Wang, IEEE/ACM TON, 2004)
- Average Session Time
 - 60 min median: Napster and Gnutella (Saroiu 2002)
- Open IP address vs. behind NAT
- Latency
 - 20% peers \leq 70ms, 20% peers \geq 280ms (Saroiu 2002)
- CPU/Memory

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.18

Aside: Network Resilience



Partial Topology

from Saroiu *et al.*, MMCN 2002

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.19

KaZaA: History

- KaZaA created by Dutch company Kazaa BV in 2001
 - FastTrack network (other clients: Morpheus, giFT,...)
 - Protocol later changed so other clients couldn't talk to it
- Very popular file sharing network >10M users(?)
- Included lots of malware (at different times):
 - Cydoor (spyware): Collects info PC's surfing habits
 - B3D (adware): Add-on which causes advertising popups
 - Altnet (adware): Distrib. network for paid "gold" files
 - The Best Offers (adware): Tracks your browsing habits and internet usage to display ads
 - InstaFinder (hijacker): Redirects URL typing errors to InstaFinder's web page instead of standard search page.
 - TopSearch (adware): Displays paid media related to searches
 - RX Toolbar (spyware): Toolbar tracks your browsing habits
 - New.net (hijacker): unofficial Top Level Domain names

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.20

KaZaA: Overview

- **"Smart" Query Flooding:**
 - Join: on startup, client contacts a "supernode" ... may at some point become one itself
 - Publish: send list of files to supernode
 - Search: send query to supernode, supernodes flood query amongst themselves.
 - Fetch: get the file directly from peer(s); can fetch simultaneously from multiple peers
- Lots of work to make connections between peers easy
 - Hole punching through different (broken) NAT implementations
- Identifying supernodes
 - High BW, always on
 - What types of PCs fall into this category?

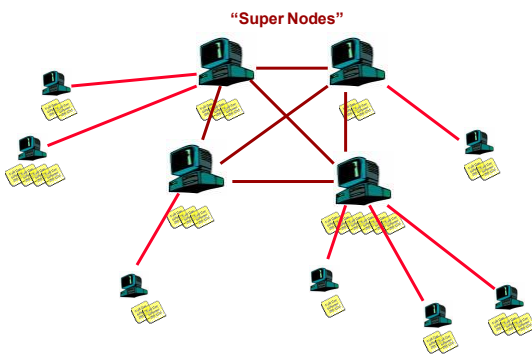
5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.21

BREAK

KaZaA: Network Design

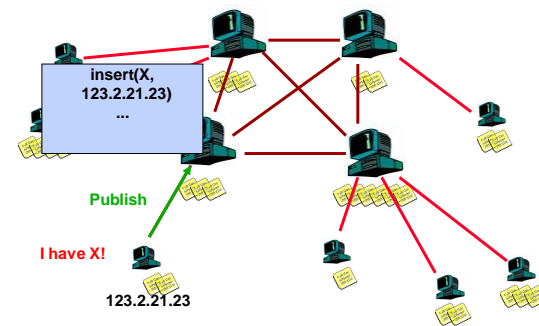


5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.23

KaZaA: File Insert

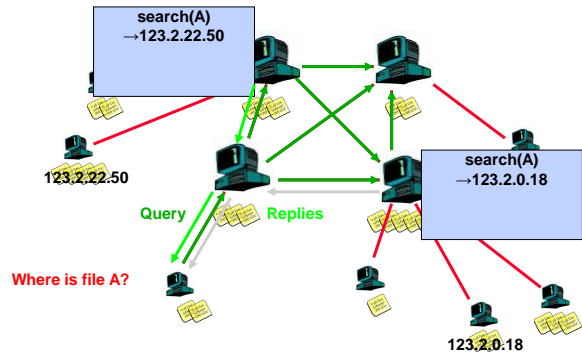


5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.24

KaZaA: File Search



5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.25

KaZaA: Discussion

- Pros:
 - Tries to take into account node heterogeneity:
 - » Bandwidth
 - » Host Computational Resources
 - » Host Availability
 - Rumored to take into account network locality
- Cons:
 - Still no real guarantees on search scope or search time
- Technical basis for Skype (P2P VoIP) and Joost (P2P video)

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.26

BitTorrent: Overview

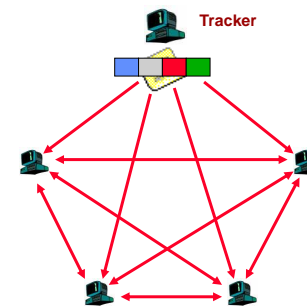
- B. Cohen designed BitTorrent in 2001
- Key Motivation:
 - Popularity exhibits temporal locality (Flash Crowds)
 - E.g., Slashdot effect, CNN on 9/11, new movie/game release
- Focused on Efficient *Fetching*, not *Searching*:
 - Distribute the *same* file to all peers
 - Single publisher, multiple downloaders
- Has some "real" publishers:
 - Blizzard Entertainment used it to distribute a game beta
 - BitTorrent company - Torrent Entertainment media catalog
- Swarming:
 - Join: contact centralized "tracker" server, get a list of peers
 - Publish: Run a tracker server
 - Search: Out-of-band (Use Google to find a tracker for the file you want)
 - Fetch: Download chunks of the file from your peers, Upload chunks you have to them

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.27

BitTorrent: Publish/Join

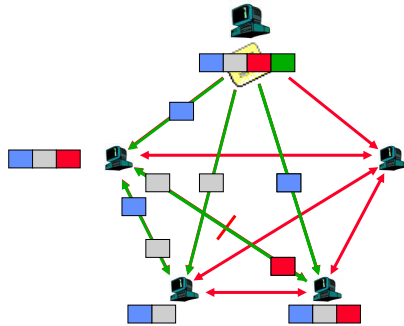


5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.28

BitTorrent: Fetch



5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.29

BitTorrent: Summary

- **Employ "Tit-for-tat" sharing strategy**
 - "I'll share with you if you share with me"
 - **Be optimistic: occasionally let freeloaders download**
 - » Otherwise no one would ever start!
 - » Also allows you to discover better peers to download from when they reciprocate
- **Pros:**
 - Works reasonably well in practice
 - Gives peers incentive to share resources; avoids freeloaders
- **Cons:**
 - Central tracker server needed to bootstrap swarm (is this really necessary?)

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.30

Conclusions

- **Napster**
 - Launched P2P
 - Centralized index
- **Gnutella:**
 - Focus is simple sharing
 - Using simple flooding
- **Kazaa/Skype/Joost**
 - More intelligent query routing
- **BitTorrent**
 - Focus on download speed, fairness in sharing
- **Want to learn more about P2P networking?**
 - Take EE122
- **Contact me if you're interested in doing undergraduate research at Berkeley or Intel Research Berkeley**

5/12/08

Joseph CS162 ©UCB Spring 2008

Lec 27.31