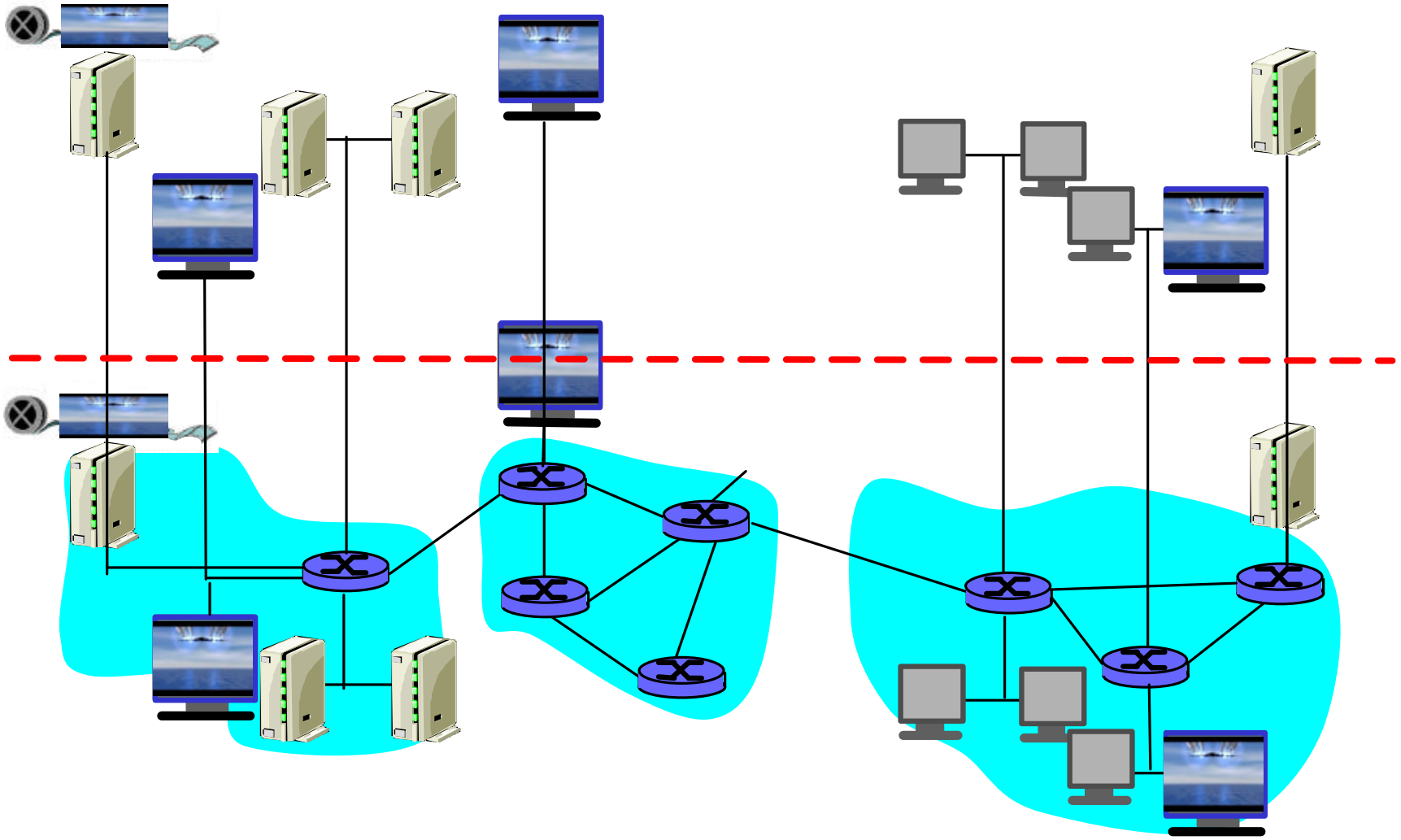
Peer-peer and Application-level Networking

CS 218 Fall 2003

Multicast Overlays
P2P applications
    Napster, Gnutella, Robust Overlay Networks
Distributed Hash Tables (DHT)
    Chord
    CAN

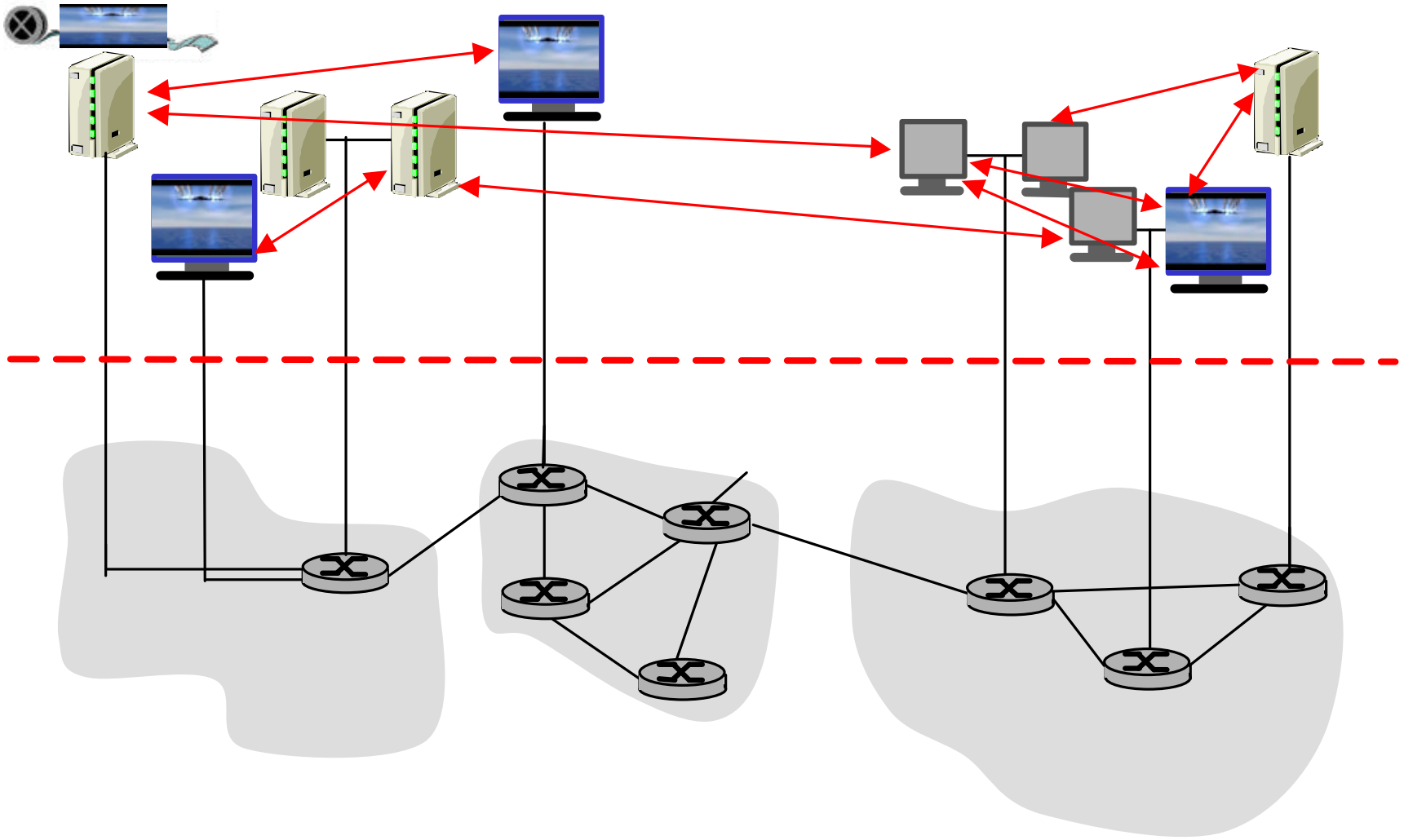
    Much of this material comes from UMASS class slides

Peer-peer networking



Peer-peer networking

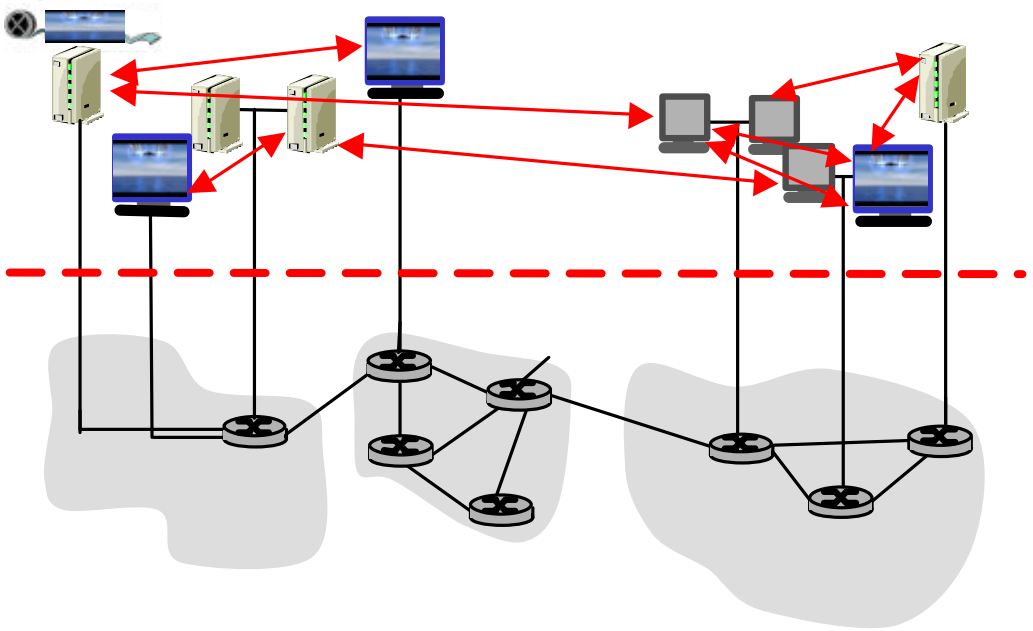
Focus at the application level



Peer-peer networking

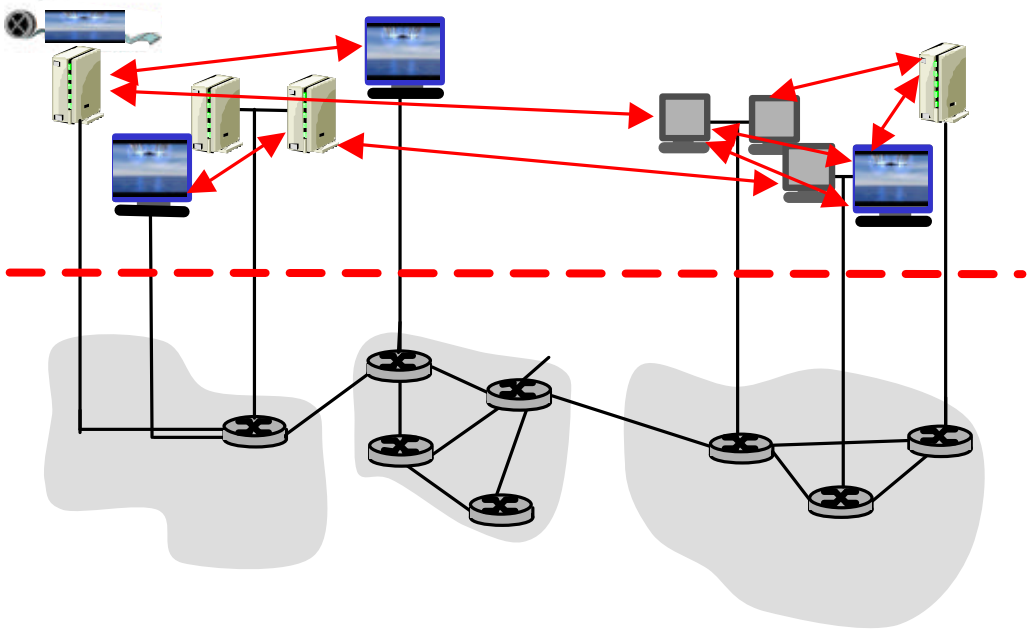
Peer-peer applications

- Napster, Gnutella, Freenet: file sharing
- ad hoc networks
- multicast overlays (e.g., video distribution)



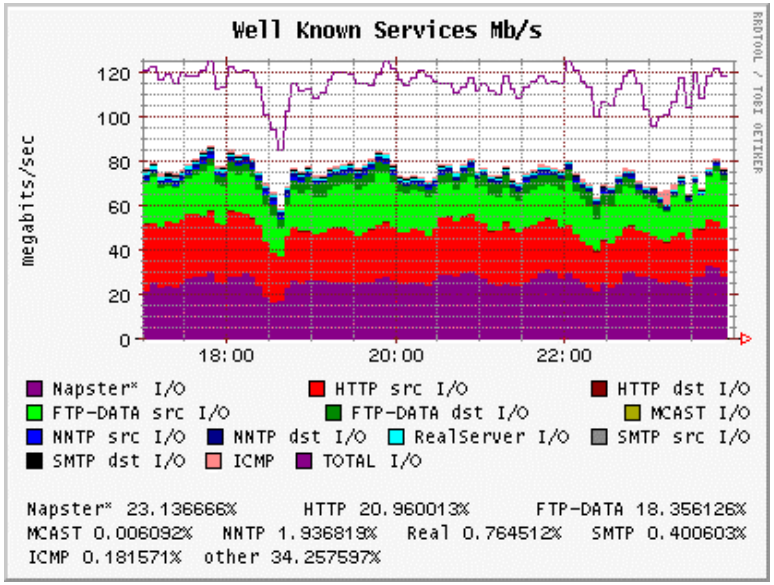
Peer-peer networking

- Q: What are the new technical challenges?
- Q: What new services/applications enabled?
- Q: Is it just "networking at the application-level"?
  - Everything old is new again?



Napster

- program for sharing files over the Internet
- a “disruptive” application/technology?
- history:
  - 5/99: Shawn Fanning (freshman, Northeasten U.) founds Napster Online music service
  - 12/99: first lawsuit
  - 3/00: 25% UWisc traffic Napster
  - 2000: est. 60M users
  - 2/01: US Circuit Court of Appeals: Napster knew users violating copyright laws
  - 7/01: # simultaneous online users: Napster 160K, Gnutella: 40K, Morpheus: 300K



Napster: how does it work

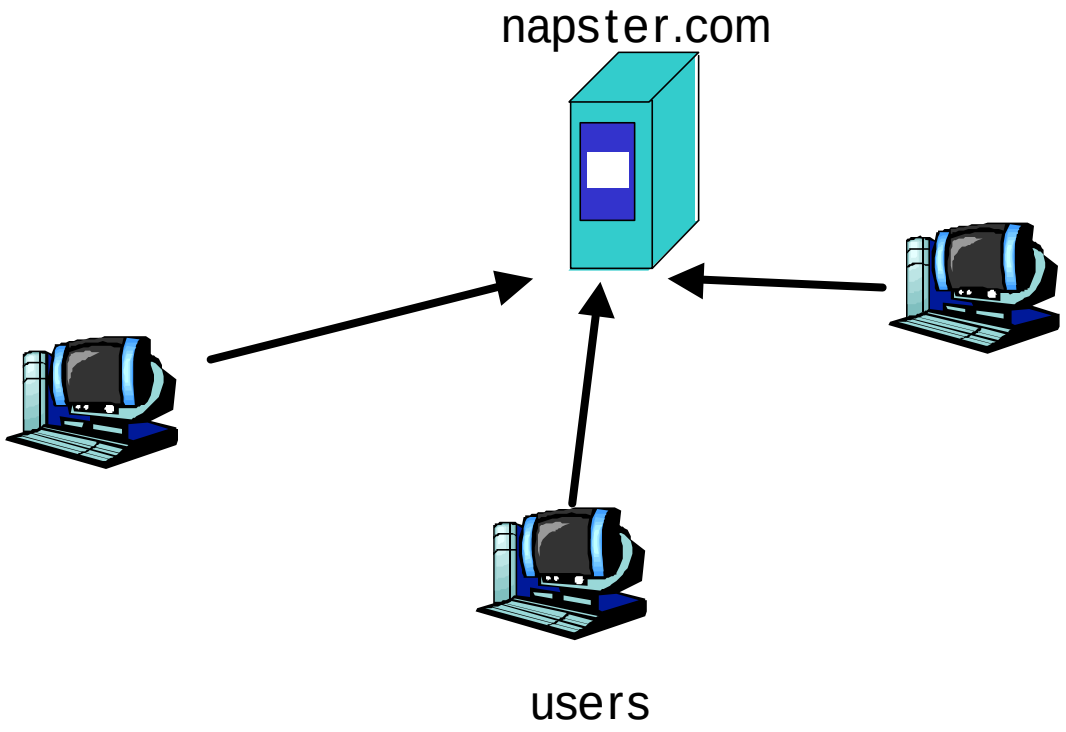
Application-level, client-server protocol over point-to-point TCP

Four steps:

- Connect to Napster server
- Upload your list of files (push) to server.
- Give server keywords to search the full list with.
- Select “best” of correct answers. (pings)

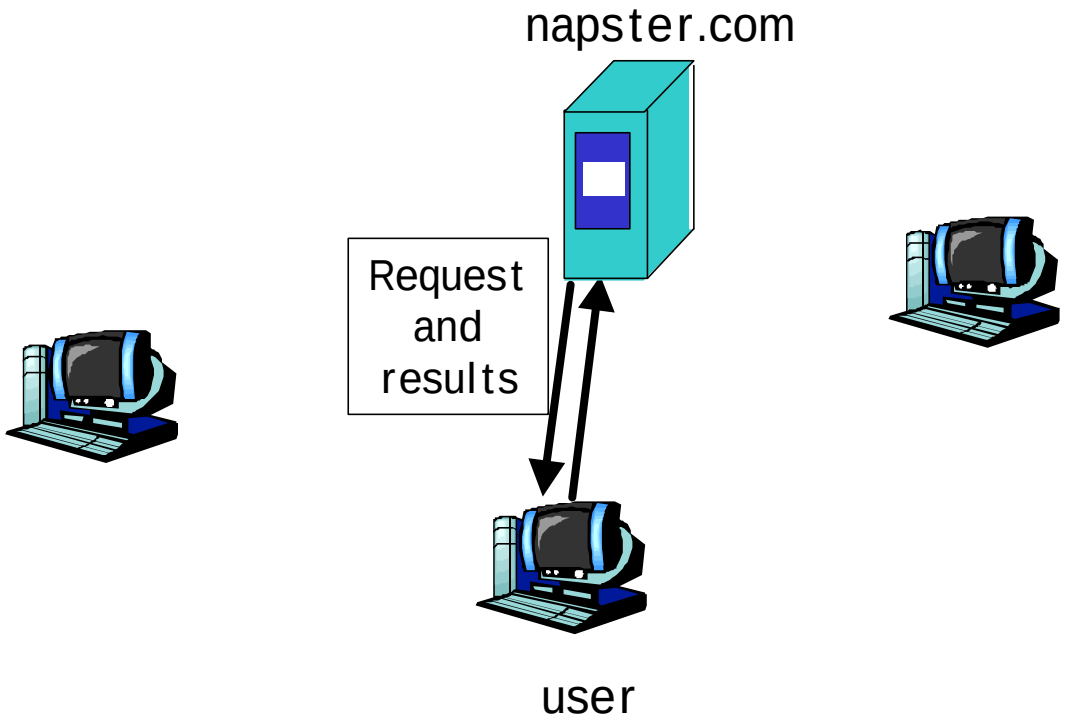
Napster

1. File list is uploaded



Napster

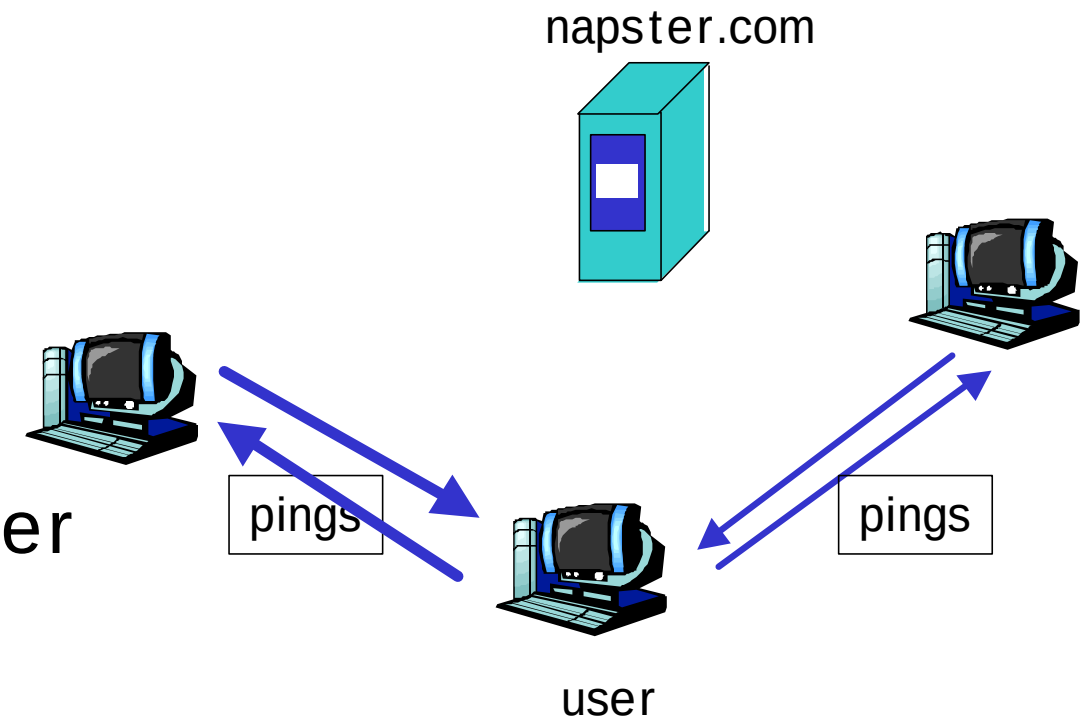
2. User requests search at server.



Napster

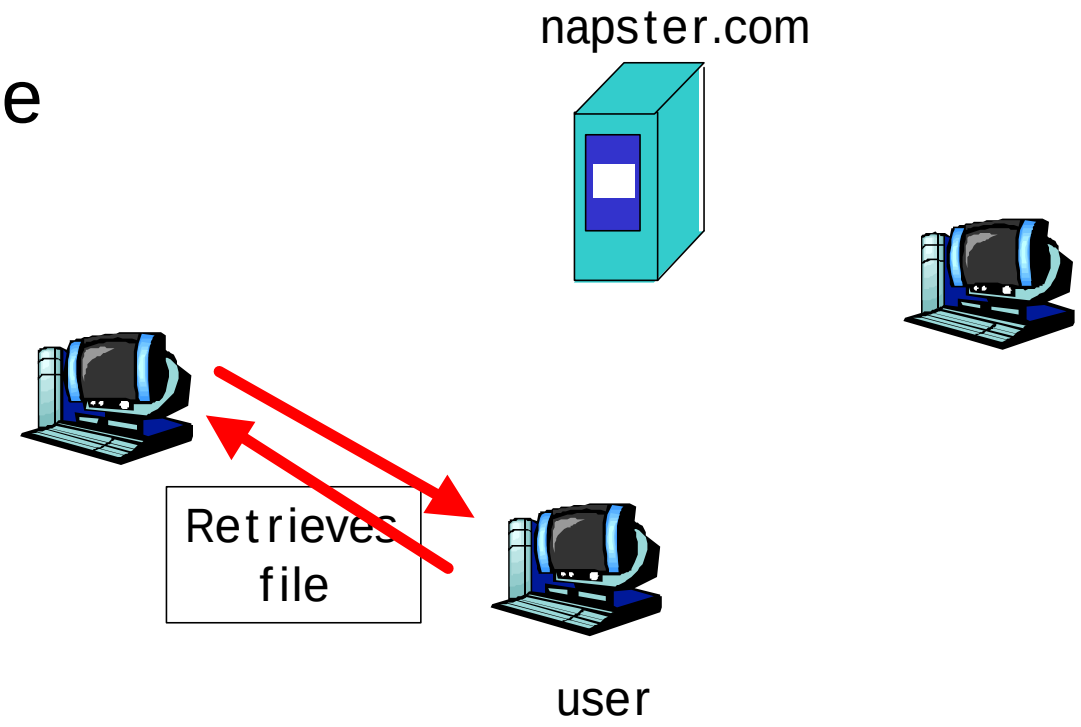
3. User pings hosts that apparently have data.

   Looks for best transfer rate.



Napster

4. User retrieves file



Napster messages

General Packet Format

[chunksize] [chunkinfo] [data...]

CHUNKSIZE:
Intel-endian 16-bit integer
size of [data...] in bytes

CHUNKINFO: (hex)
Intel-endian 16-bit integer.

00 - login rejected
02 - login requested
03 - login accepted
0D - challenge? (nuprin1715)
2D - added to hotlist
2E - browse error (user isn't online!)
2F - user offline

5B - whois query
5C - whois result
5D - whois: user is offline!
69 - list all channels
6A - channel info
90 - join channel
91 - leave channel
.....

From: http://david.weekly.org/code/napster.php3

Napster: requesting a file

SENT to server (after logging in to server)
2A 00 CB 00 username
"C:\MP3\REM - Everybody Hurts.mp3"

RECEIVED
5D 00 CC 00 username
2965119704 (IP-address backward-form = A.B.C.D)
6699 (port)
"C:\MP3\REM - Everybody Hurts.mp3" (song)
(32-byte checksum)
(line speed)

[connect to A.B.C.D:6699]

RECEIVED from client
31 00 00 00 00 00

SENT to client
GET

RECEIVED from client
00 00 00 00 00 00

SENT to client
Myusername
"C:\MP3\REM - Everybody Hurts.mp3"
0 (port to connect to)

RECEIVED from client
(size in bytes)

SENT to server
00 00 DD 00 (give go-ahead thru server)

RECEIVED from client
[DATA]

Napster: architecture notes

- centralized server:
  - single logical point of failure
  - can load balance among servers using DNS rotation
  - potential for congestion
  - Napster “in control” (freedom is an illusion)
- no security:
  - passwords in plain text
  - no authentication
  - no anonymity

Gnutella

- peer-to-peer networking: applications connect to peer applications
- focus: decentralized method of searching for files
- each application instance serves to:
  - store selected files
  - route queries (file searches) from and to its neighboring peers
  - respond to queries (serve file) if file stored locally
- Gnutella history:
  - 3/14/00: release by AOL, almost immediately withdrawn
  - too late: several thousands of users on Gnutella as of now
  - many iterations to fix poor initial design (poor design turned many people off)

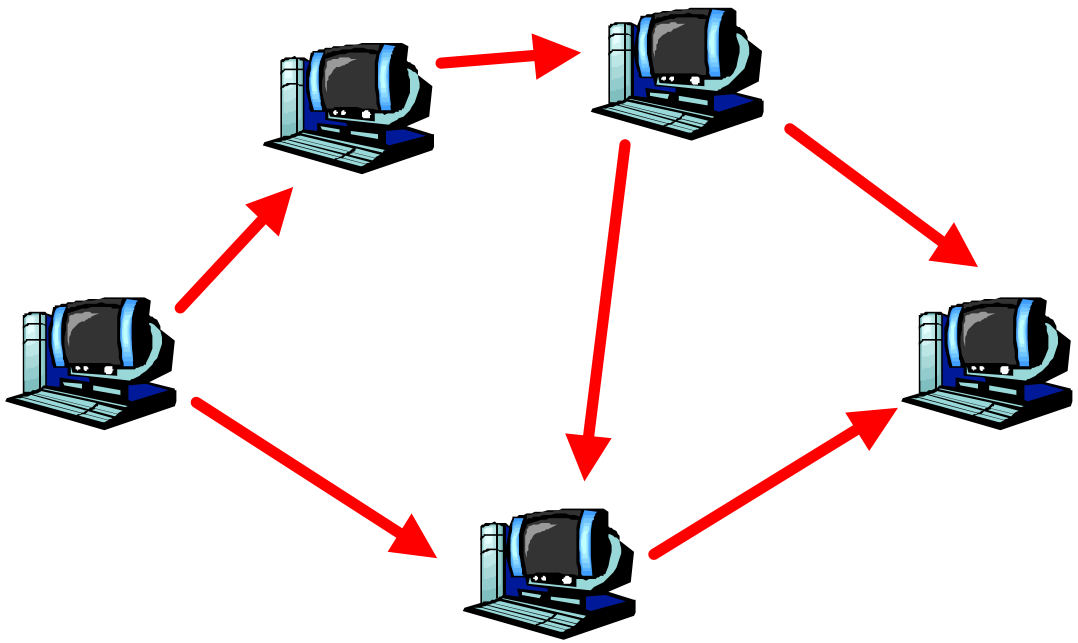
Gnutella: how it works

Searching by flooding:

- If you don't have the file you want, query 7 of your partners.
- If they don't have it, they contact 7 of their partners, for a maximum hop count of 10.
- Requests are flooded, but there is no tree structure.
- No looping but packets may be received twice.
- Reverse path forwarding(?)

Note: Play gnutella animation at:
http://www.limewire.com/index.jsp/p2p

Flooding in Gnutella: loop prevention



Seen already list: “A”

Gnutella message format

- Message ID: 16 bytes (yes bytes)
- FunctionID: 1 byte indicating
  - 00 ping: used to probe gnutella network for hosts
  - 01 pong: used to reply to ping, return # files shared
  - 80 query: search string, and desired minimum bandwidth
  - 81: query hit: indicating matches to 80:query, my IP address/port, available bandwidth
- RemainingTTL: decremented at each peer to prevent TTL-scoped flooding
- HopsTaken: number of peer visited so far by this message
- DataLength: length of data field

Gnutella: initial problems and fixes

- Freeloading: WWW sites offering search/retrieval from Gnutella network without providing file sharing or query routing.
  - Block file-serving to browser-based non-file-sharing users
- Prematurely terminated downloads:
  - long download times over modems
  - modem users run gnutella peer only briefly (Napster problem also!) or any users becomes overloaded
  - fix: peer can reply "I have it, but I am busy. Try again later"
  - late 2000: only 10% of downloads succeed
  - 2001: more than 25% downloads successful (is this success or failure?)

Gnutella: initial problems and fixes (more)

- 2000: avg size of reachable network only 400-800 hosts. Why so smalll?
  - modem users: not enough bandwidth to provide search routing capabilities: routing black holes
- Fix: create peer hierarchy based on capabilities
  - previously: all peers identical, most modem blackholes
  - connection preferencing:
    - favors routing to well-connected peers
    - favors reply to clients that themselves serve large number of files: prevent freeloading

Anonymous?

- Not anymore than it's scalable.
- The person you are getting the file from knows who you are. That's not anonymous.
- Other protocols exist where the owner of the files doesn't know the requester.
- Peer-to-peer anonymity exists.

Gnutella Discussion:

- Architectural lessons learned?
  - ..
  - ..
  - ..
  - ..
- Do Gnutella's goals seem familiar? Does it work better than say squid or summary cache?
- anonymity and security?
- Other?
- Good source for technical info/open questions:
  - http://www.limewire.com/index.jsp/tech_papers