Computer Science 425

Distributed Systems

CS 425 / ECE 428

Fall 2013

Indranil Gupta (Indy)

September 26, 2013

Lecture 10

Peer-to-peer Systems I

Reading: Gnutella paper on website

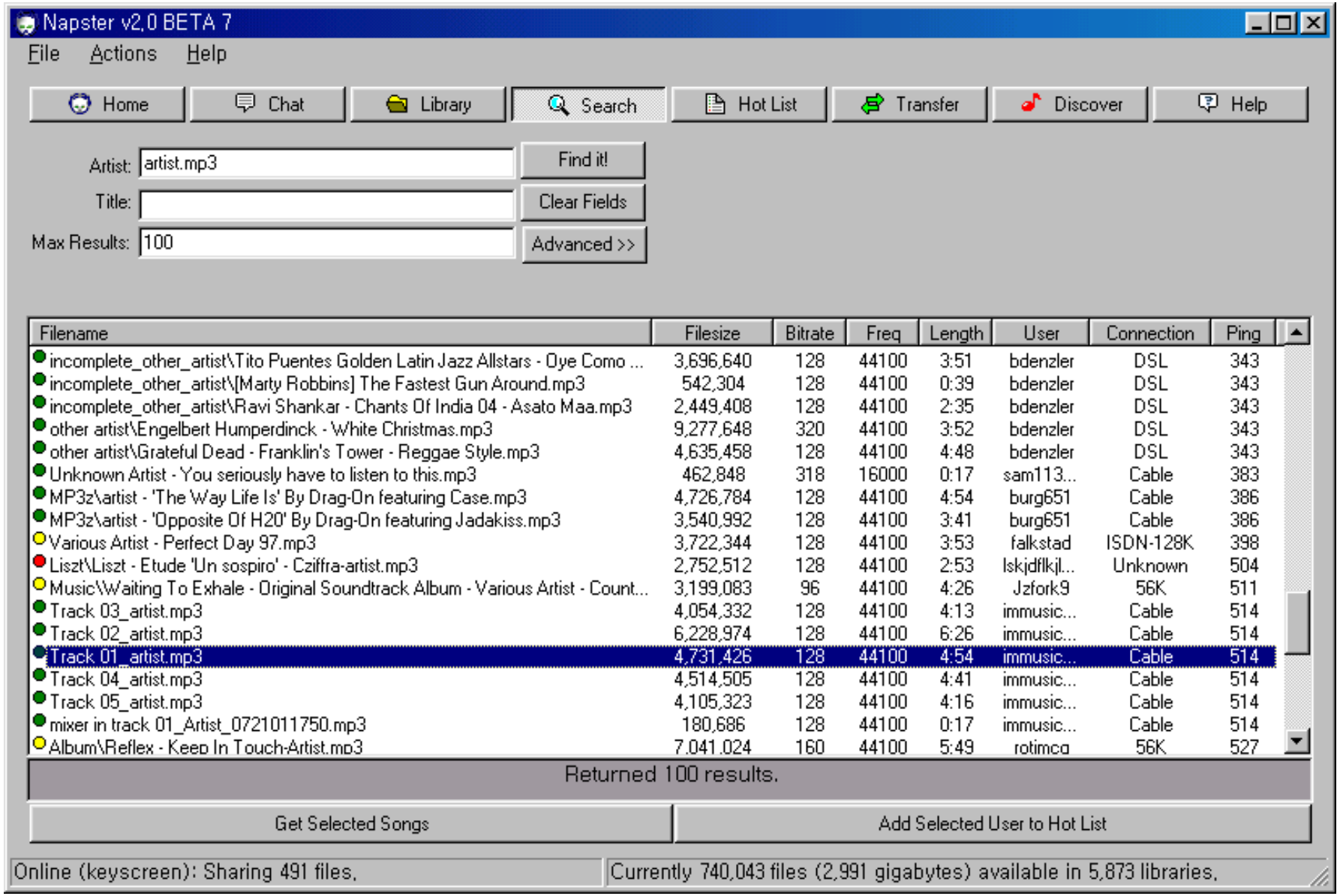
Why Study Peer to peer (P2P) systems?

- To understand how they work
- To understand the **techniques and principles** within them
- To modify, adapt, reuse these techniques and principles in other related areas
  - Cloud computing: key-value stores borrow heavily from p2p systems
  - To build your own p2p system
- To grow the body of knowledge about distributed systems

Some Questions

- Why do people get together?
  - to share information
  - to share and exchange resources they have
    - books, class notes, experiences, videos, music cd's
- How can computers help people
  - find information
  - find resources
  - exchange and share resources

- Existing technologies: The Web!
  - Search engines
  - Forums: chat rooms, blogs, ebay
  - Online business
- But, the web is heavy weight if you want specific resources: say a Beatles' song "PennyLane"
- A search engine will give you their bio, lyrics, chords, articles on them, and then perhaps the mp3
- But you want only the song, nothing else!
- **If you can find a peer who has a copy of the Beatles song (mp3), perhaps in exchange for your UIUC Homeocoming videos, that would be great!**
  - **Napster: a solution light weight that was lighter than the Web**


Napster v2.0 BETA 7
File Actions Help
Home Chat Library Search Hot List Transfer Discover Help
Artist: artist.mp3
Title:
Max Results: 100
Find it!
Clear Fields
Advanced >>
Filename Filesize Bitrate Freq Length User Connection Ping
incomplete_other_artist\Tito Puentes Golden Latin Jazz Allstars - Oye Como ... 3,696,640 128 44100 3:51 bdenzler DSL 343
incomplete_other_artist\[Marty Robbins] The Fastest Gun Around.mp3 542,304 128 44100 0:39 bdenzler DSL 343
incomplete_other_artist\Ravi Shankar - Chants Of India 04 - Asato Maa.mp3 2,449,408 128 44100 2:35 bdenzler DSL 343
other artist\Engelbert Humperdinck - White Christmas.mp3 9,277,648 320 44100 3:52 bdenzler DSL 343
other artist\Grateful Dead - Franklin's Tower - Reggae Style.mp3 4,635,458 128 44100 4:48 bdenzler DSL 343
Unknown Artist - You seriously have to listen to this.mp3 462,848 318 16000 0:17 sam113... Cable 383
MP3z\artist - 'The Way Life Is' By Drag-On featuring Case.mp3 4,726,784 128 44100 4:54 burg651 Cable 386
MP3z\artist - 'Opposite Of H20' By Drag-On featuring Jadakiss.mp3 3,540,992 128 44100 3:41 burg651 Cable 386
Various Artist - Perfect Day 97.mp3 3,722,344 128 44100 3:53 falkstad ISDN-128K 398
Liszt\Liszt - Etude 'Un sospiro' - Cziffra-artist.mp3 2,752,512 128 44100 2:53 lskjdflkjl... Unknown 504
Music\Waiting To Exhale - Original Soundtrack Album - Various Artist - Count... 3,199,083 96 44100 4:26 Jzfork9 56K 511
Track 03_artist.mp3 4,054,332 128 44100 4:13 immusic... Cable 514
Track 02_artist.mp3 6,228,974 128 44100 6:26 immusic... Cable 514
Track 01_artist.mp3 4,731,426 128 44100 4:54 immusic... Cable 514
Track 04_artist.mp3 4,514,505 128 44100 4:41 immusic... Cable 514
Track 05_artist.mp3 4,105,323 128 44100 4:16 immusic... Cable 514
mixer in track 01_Artist_0721011750.mp3 180,686 128 44100 0:17 immusic... Cable 514
Album\Reflex - Keep In Touch-Artist.mp3 7,041,024 160 44100 5:49 rotimco 56K 527
Returned 100 results.
Get Selected Songs
Add Selected User to Hot List
Online (keyscreen): Sharing 491 files.
Currently 740,043 files (2,991 gigabytes) available in 5,873 libraries.

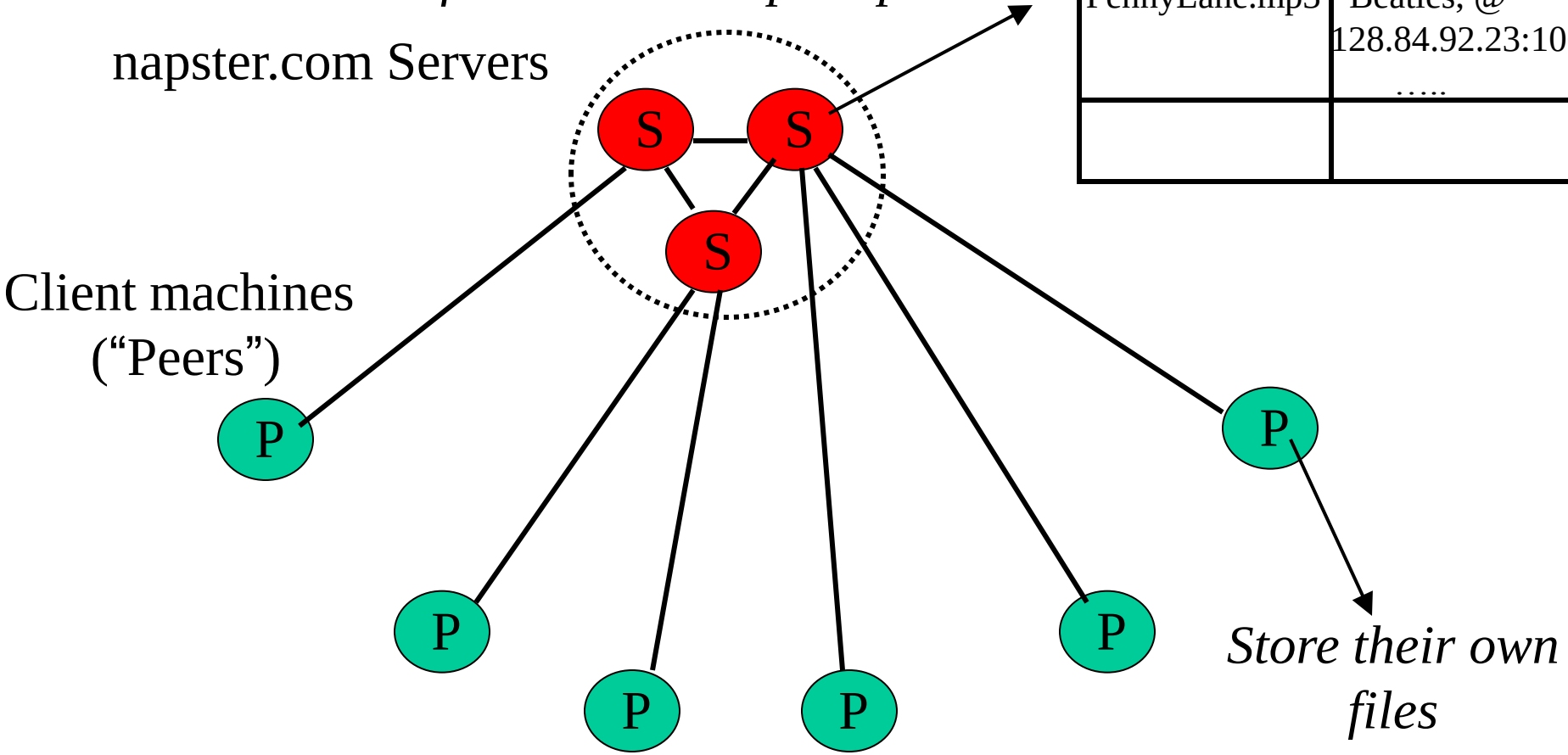
A Brief History

- [6/99] Shawn Fanning (freshman Northeastern U.) releases Napster online music service
- [12/99] RIAA sues Napster, asking $100K per download
- [3/00] 25% UWisc traffic Napster, many universities ban it
- [00] 60M users
- [2/01] US Federal Appeals Court: users violating copyright laws, Napster is abetting this
- [9/01] Napster decides to run paid service, pay % to songwriters and music companies
- [Today] Napster protocol is open, people free to develop opennap clients and servers
  http://opennap.sourceforge.net

Napster Structure

Filename	Info about
PennyLane.mp3	Beatles, @ 128.84.92.23:1006 …..

Store a directory, i.e., filenames with peer pointers

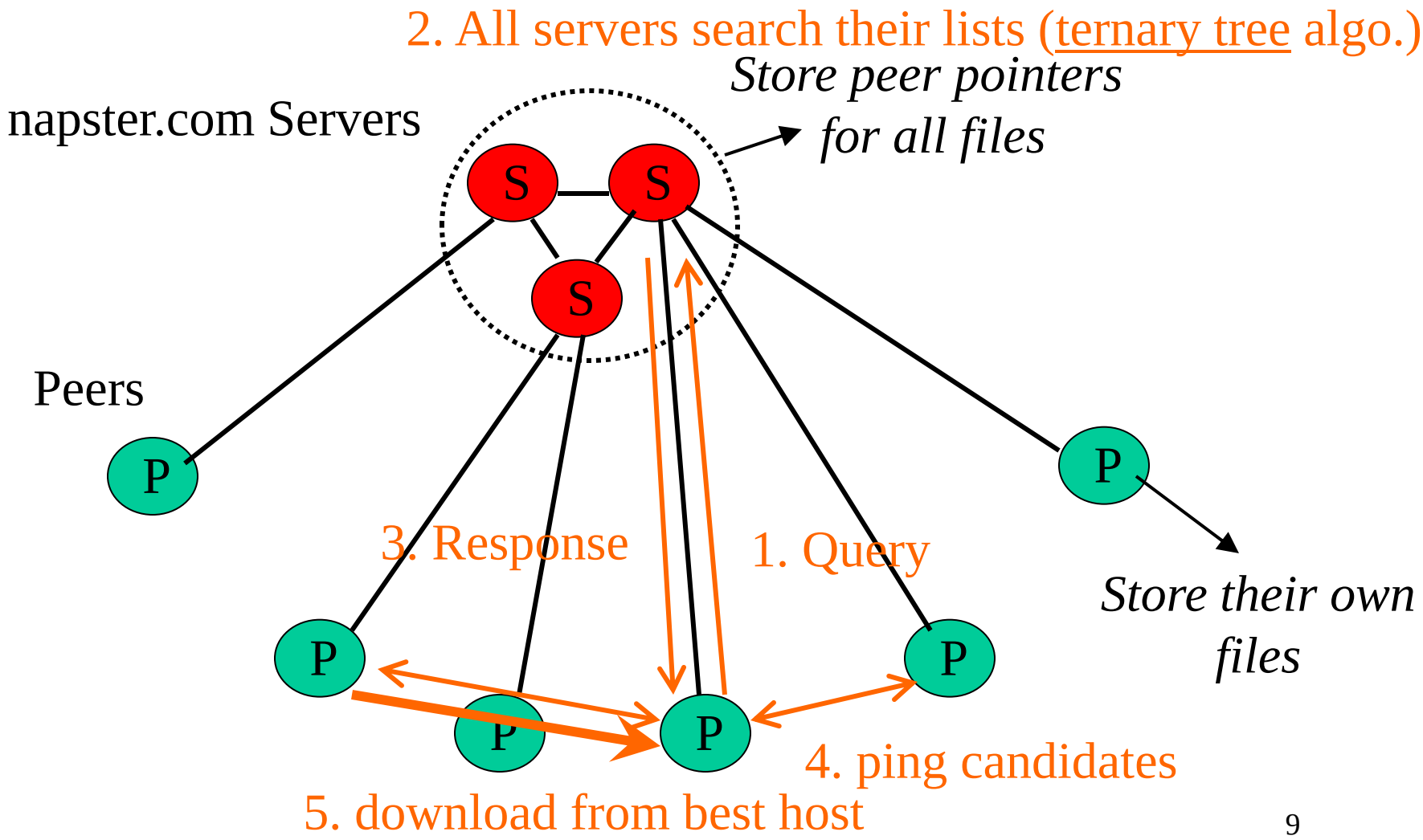


Napster Operations

Client

- Connect to a Napster server
- Upload list of music files that you want to share
  - Server maintains list of <filename, ip_address, portnum> tuples. Server stores no files.
- Search
  - Send server keywords to search with
  - (Server searches its list with the keywords)
  - Server returns a list of hosts - <ip_address, portnum> tuples - to client
  - Client pings each host in the list to find transfer rates
  - Client fetches file from best host
- All communication uses TCP

Napster Search



Problems

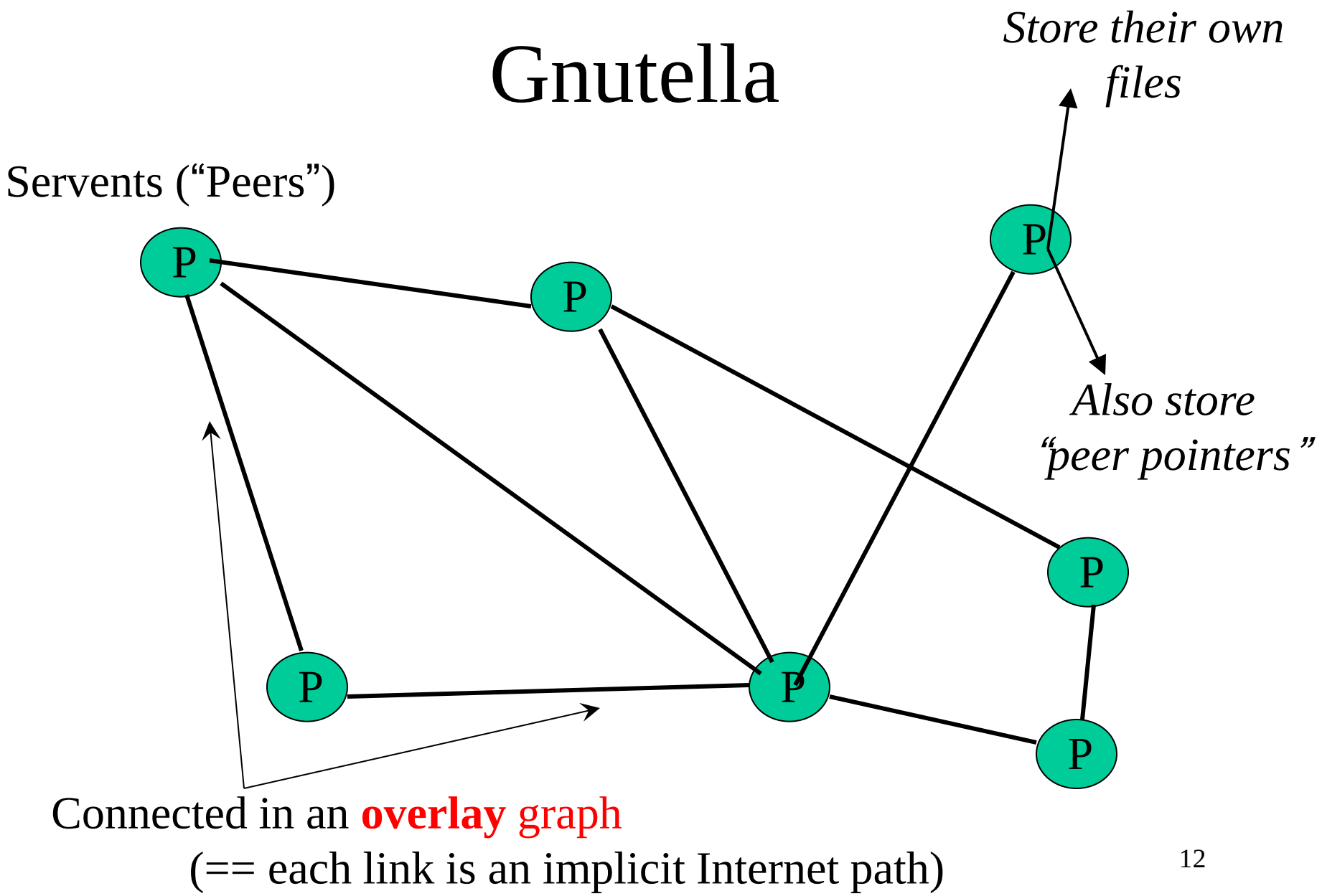
- Centralized server a source of congestion
- Centralized server single point of failure
- No security: plaintext messages and passwds
- Courts declared napster.com responsible for users' copyright violation
  - "Indirect infringement"

Gnutella

- Eliminate the servers
- Client machines search and retrieve amongst themselves
- Clients act as servers too, called **servents**
- [3/00] release by AOL, 88K users by 3/03
- Original design underwent several modifications
- Available as an open protocol today

http://www.limewire.com

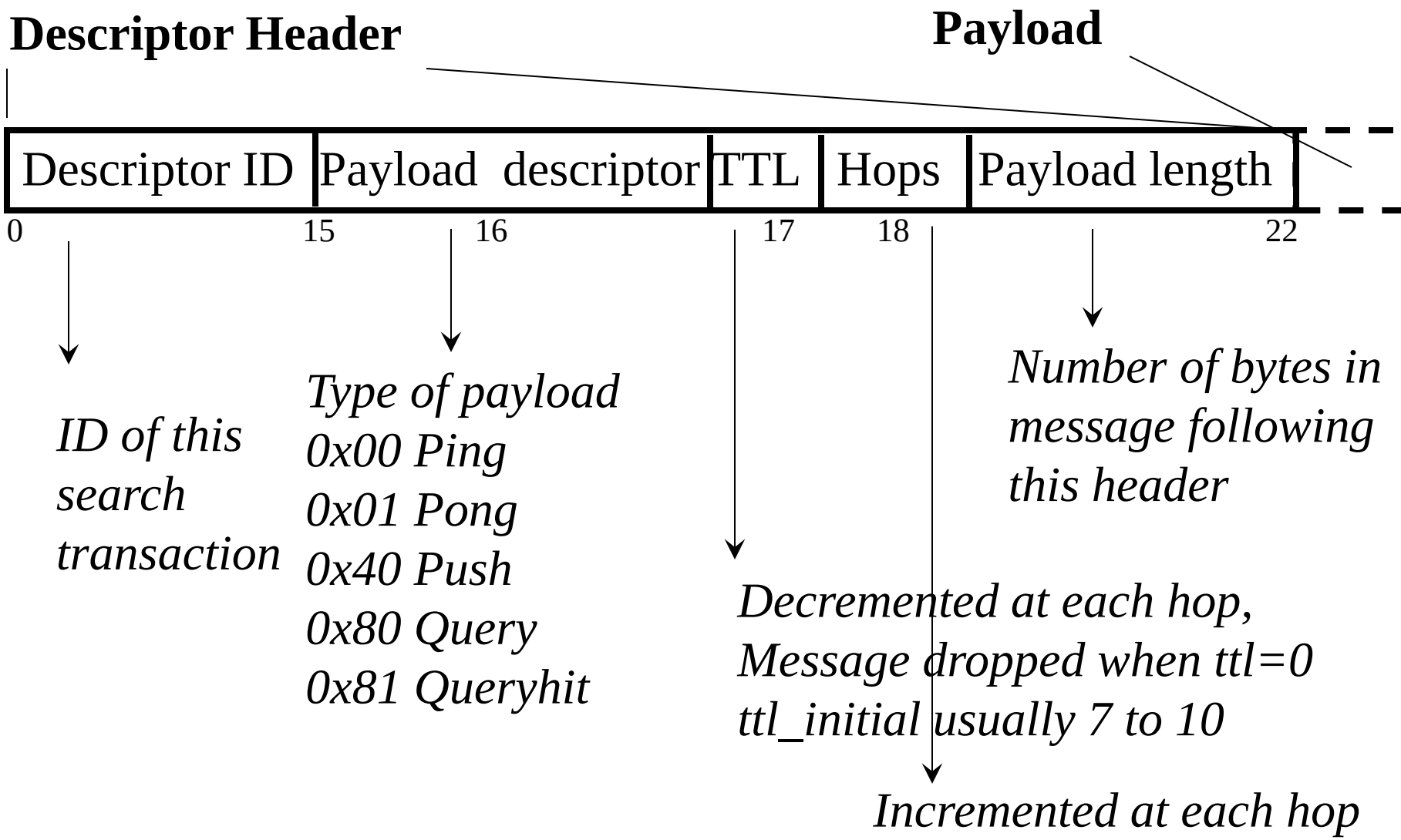
Gnutella


Servents (“Peers”)
Store their own files
Also store “peer pointers”
P
P
P
P
P
P
P
Connected in an **overlay** graph
(== each link is an implicit Internet path)

How do I search for my Beatles file?

- Gnutella *routes* different messages within the overlay graph
- Gnutella protocol has 5 main message types
  - Query (search)
  - QueryHit (response to query)
  - Ping (to probe network for other peers)
  - Pong (reply to ping, contains address of another peer)
  - Push (used to initiate file transfer)
- We'll go into the message structure and protocol now

  (note: all fields except IP address are in little-endian format)



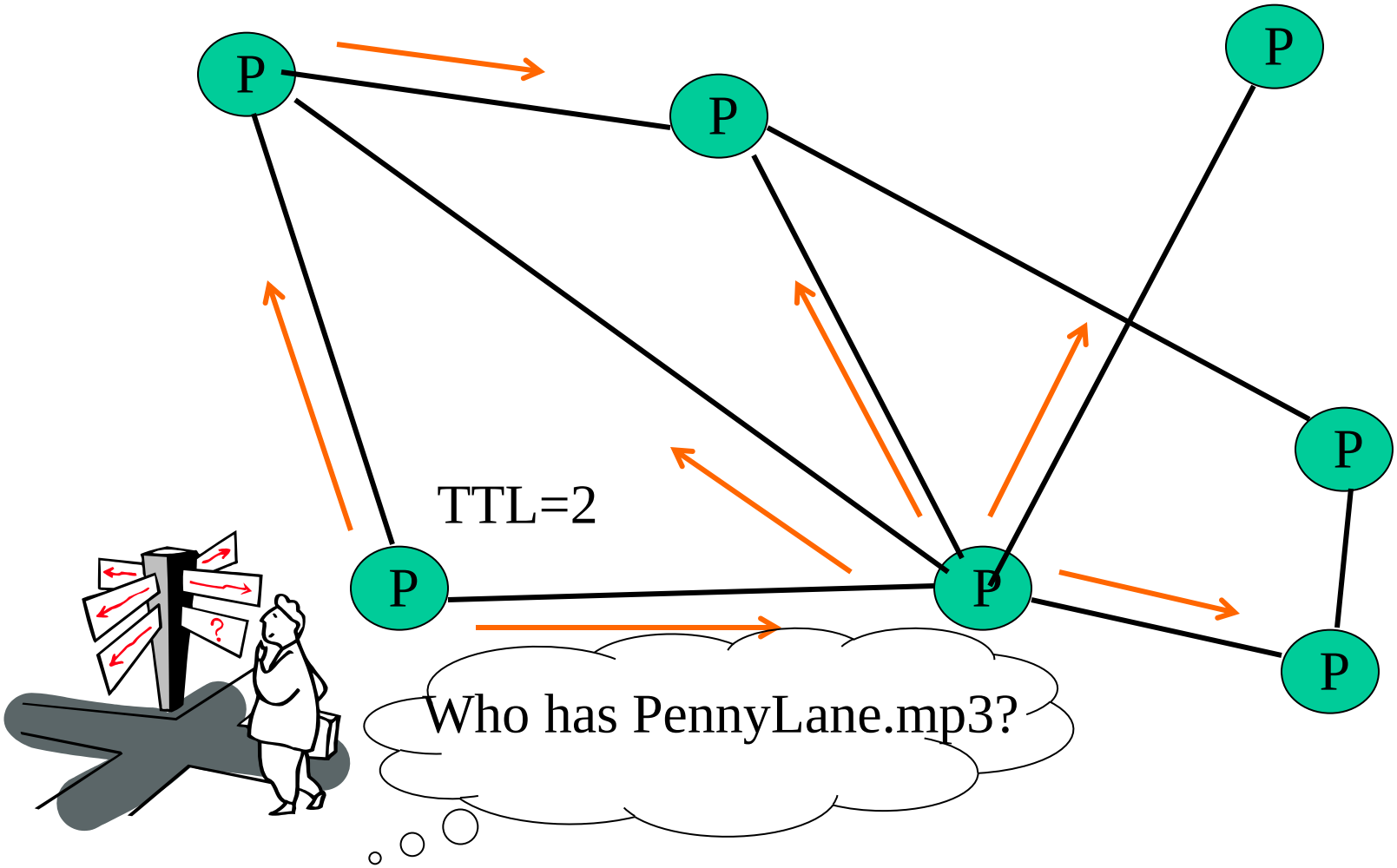
Gnutella Message Header Format

Query (0x80)

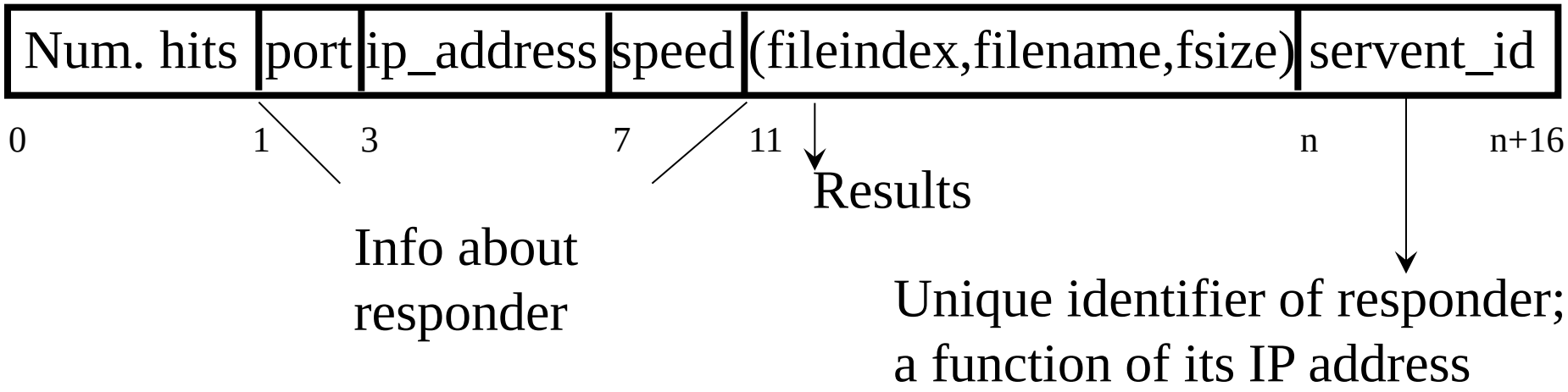
Payload Format in Gnutella Query Message

Gnutella Search

Query's flooded out, ttl-restricted, forwarded only once



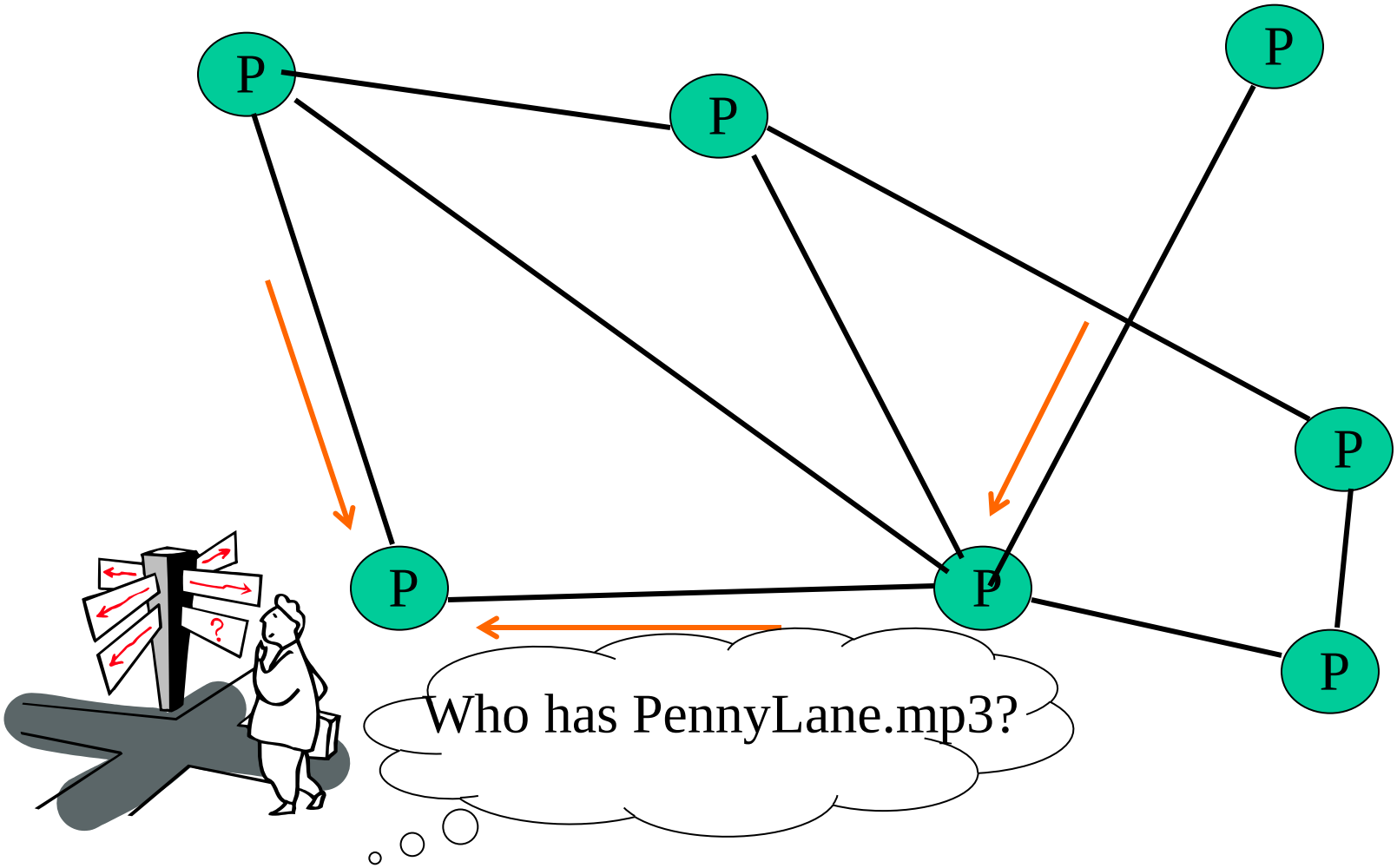
QueryHit (0x81) : successful result to a query



Payload Format in Gnutella Query Reply Message

Gnutella Search

Successful results QueryHit's routed on reverse path



Avoiding excessive traffic

To avoid duplicate transmissions, each peer maintains a list of recently received messages

- Query forwarded to all neighbors except peer from which received
- Each Query (identified by DescriptorID) forwarded only once
- QueryHit routed back only to peer from which Query received with same DescriptorID
  - If neighbor does not exist anymore, drop QueryHit
- Duplicates with same DescriptorID and Payload descriptor (msg type) are dropped
- QueryHit with DescriptorID for which Query was not seen is dropped

After receiving QueryHit messages

- Requestor chooses “best” QueryHit responder
  - Initiates HTTP request directly to responder’s ip+port

```
GET /get/<File Index>/<File Name>/HTTP/1.0\r\n
Connection: Keep-Alive\r\n
Range: bytes=0-\r\n
User-Agent: Gnutella\r\n
\r\n
```

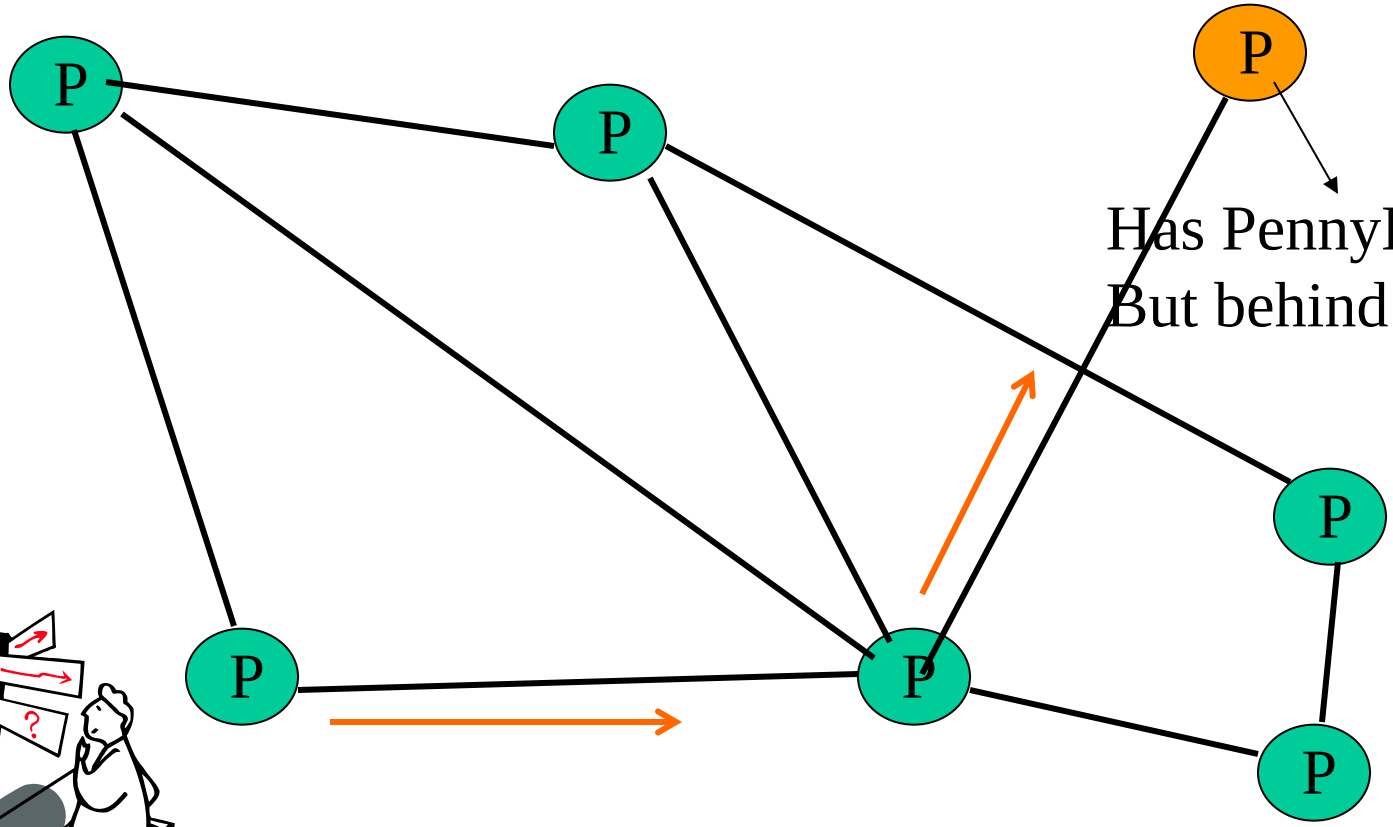
- Responder then replies following start message, followed by packets containing file:

```
HTTP 200 OK\r\n
Server: Gnutella\r\n
Content-type:application/binary\r\n
Content-length: 1024 \r\n
\r\n
```

- HTTP is the file transfer protocol. Why?
- Why the “range” field in the GET request?
- What if responder is behind firewall that disallows incoming connections?

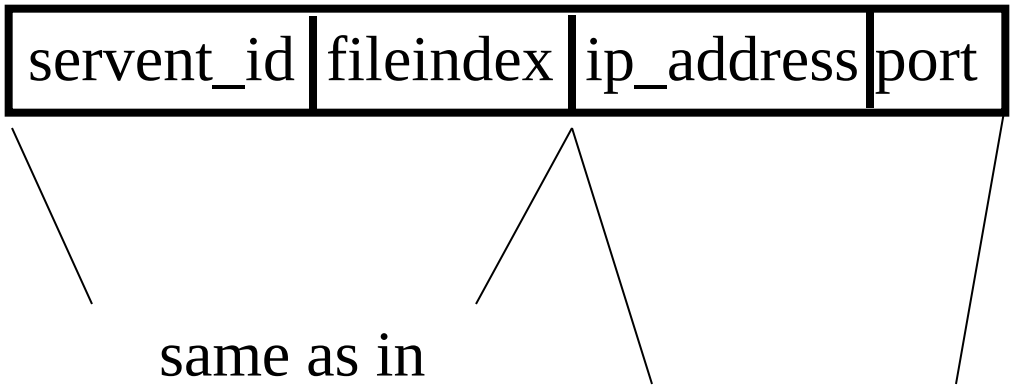
Dealing with Firewalls

Requestor sends Push to responder asking for file transfer



(Why is the Push routed and not sent directly?)

Push (0x40)


servent_id
fileindex
ip_address
port
same as in
received QueryHit
Address at which
requestor can accept
incoming connections

- Responder establishes a TCP connection at ip_address, port specified. Sends

  GIV <File Index>:<Servent Identifier>/<File Name>\n\n

- Requestor then sends GET to responder (as before) and file is transferred

- What if requestor is behind firewall too?
  - Gnutella gives up
  - Can you think of an alternative solution?

Ping-Pong

Ping (0x00)

no payload

Pong (0x01)

Port	ip_address	Num. files shared	Num. KB shared

- P2P systems have churn – peers continuously joining, leaving, and failing
- Peers initiate Ping's periodically
- Ping's flooded out like Query's, Pong's routed along reverse path (like QueryHit's)

- Pong replies used to update set of neighboring peers
  - to keep neighbor lists fresh in spite of churn

Gnutella Summary

- No servers
- Peers/servents maintain “neighbors”, this forms an overlay graph
- Peers store their own files
- Queries flooded out, ttl restricted
- QueryHit (replies) reverse path routed
- Supports file transfer through firewalls
- Periodic Ping-pong to continuously refresh neighbor lists
  - List size specified by user at peer : heterogeneity means some peers may have more neighbors
  - Gnutella found to follow **power law** distribution:

$$P(\#\text{links} = L) \sim L^{-k} \qquad (k \text{ is a constant})$$

Problems

- Ping/Pong constituted 50% traffic
  - Solution: Multiplex, *cache* and reduce frequency of pings/pongs
- Repeated searches with same keywords
  - Solution: *Cache* Query, QueryHit messages
- Modem-connected hosts do not have enough bandwidth for passing Gnutella traffic
  - Solution: use a central server to act as proxy for such peers
  - Another solution:
    - ➔FastTrack System (in a few slides)

Problems (contd.)

- Large number of *freeloaders*
  - 70% of users in 2000 were freeloaders
  - Only download files, never upload own files
  - Endemic to all p2p systems in deployment
- Flooding causes excessive traffic
  - Is there some way of maintaining meta-information about peers that leads to more intelligent routing?

➔ Structured Peer-to-peer systems

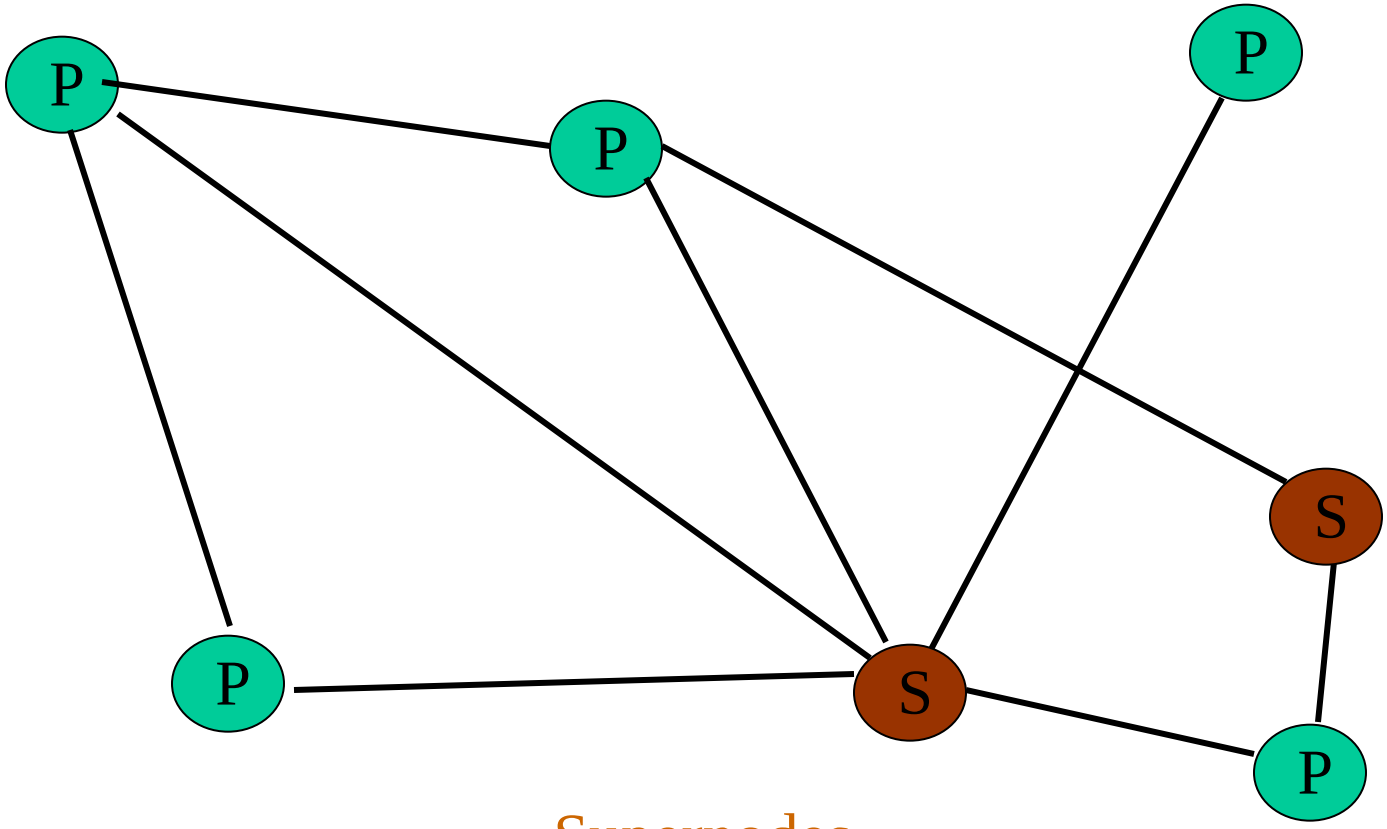
e.g., Chord System (next lecture)

FastTrack

- Hybrid between Gnutella and Napster
- Takes advantage of “healthier” participants in the system
- Underlying technology in Kazaa, KazaaLite, Grokster
- Proprietary protocol, but some details available
- Like Gnutella, but with some peers designated as *supernodes*

A FastTrack-like System

Peers

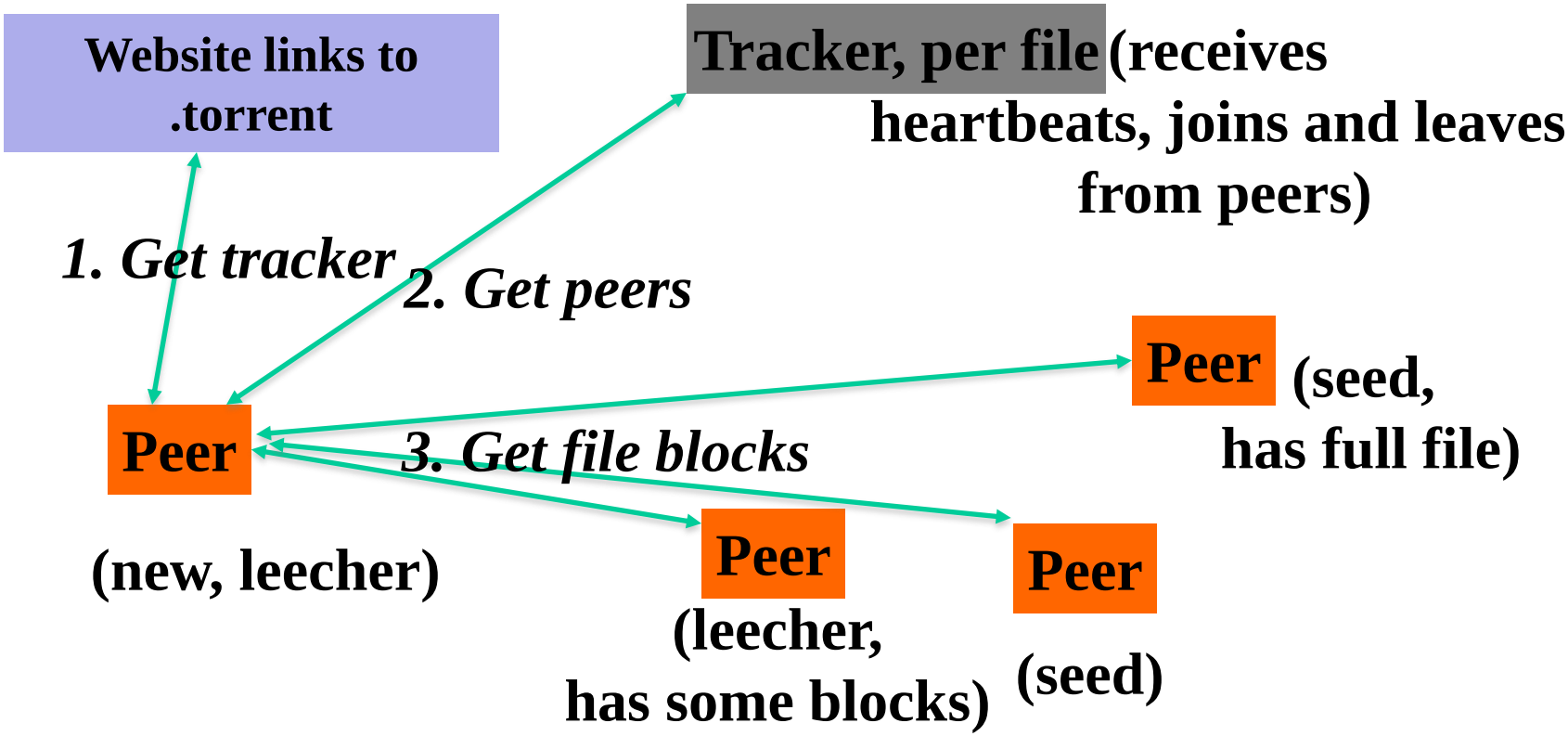


Supernodes

FastTrack (contd.)

- A supernode stores a directory listing (<filename,peer pointer>), similar to Napster servers
- A peer searches by contacting a nearby supernode
- Supernode membership changes over time
- Any peer can become (and stay) a supernode, provided it has earned enough *reputation*
  - Kazaalite: participation level (=reputation) of a user between 0 and 1000. Initially 10, then affected by length of periods of connectivity and total number of uploads.

BitTorrent – A Quick Overview



BitTorrent – A Quick Overview (2)

- File split into blocks (32 KB – 256 KB)
- Download Local Rarest First block policy: prefer early download of blocks that are least replicated among neighbors
  - Exception: New node allowed to pick one random neighbor: helps in bootstrapping
- Tit for tat bandwidth usage: Provide blocks to neighbors that provided it the best download rates
  - Incentive for nodes to provide good download rates
  - Both Leechers and Seeds
- Choking: Limit number of neighbors to which concurrent uploads <= a number (5), i.e., the "best" neighbors
  - Everyone else choked
  - Periodically re-evaluate this set (e.g., 10 s)
  - Optimistic unchoke: periodically (e.g., ~30 s), unchoke a random neigbhor – helps keep unchoked set fresh

Wrap-up Notes

Applies to all p2p systems

- How does a peer join the system
  - Send an http request to well-known url for that P2P service - `http://www.myp2pservice.com`
  - Message routed (after DNS lookup) to a well known server which then initializes new peers' neighbor table
  - Server only maintains a partial list of online clients

Summary

- Napster: protocol overview, more details available on webpage
- Gnutella protocol
- FastTrack protocol
- BitTorrent
- Protocols continually evolving, software for new clients and servers conforming to respective protocols: developer forums at
  - Napster: http://opennap.sourceforge.net
  - Gnutella: http://www.limewire.com
- Others
  - Peer to peer working groups: http://p2p.internet2.edu

For Next Lecture

- Read “Chord” paper from website
  - Sections 1-4, 6-7

- MP2
  - By now you should have a design for, and have started coding the failure detector
- Blue Waters Datacenter Tour!
  - On Oct 9th. Watch Piazza for signup sheet.