CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

P2P SYSTEMS

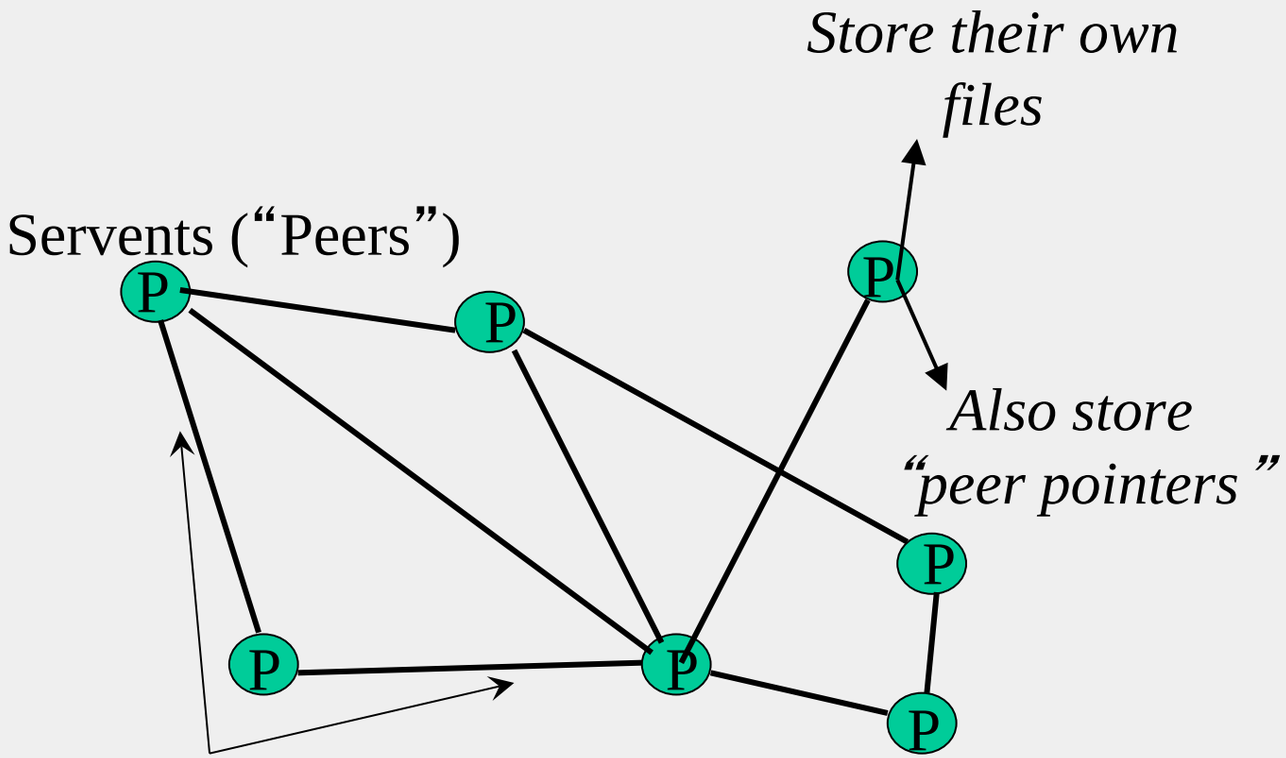
Lecture C

GNUTELLA

Gnutella

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- [3/00] release by AOL, immediately withdrawn, but 88K users by 3/03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

Gnutella



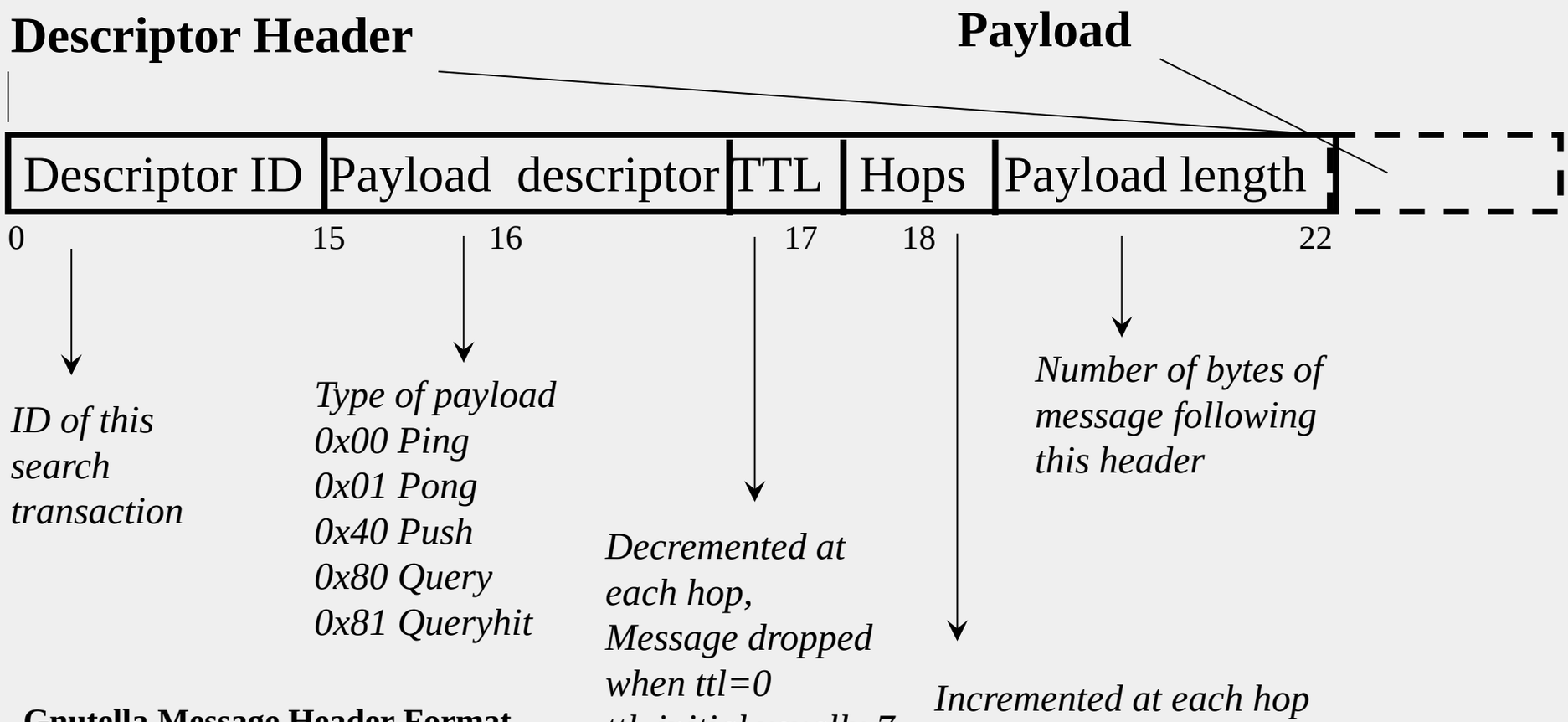
Connected in an **overlay** graph

(== 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
  - All fields except IP address are in little-endian format
  - Ox12345678 stored as 0x78 in lowest address byte, then 0x56 in next higher address, and so on.

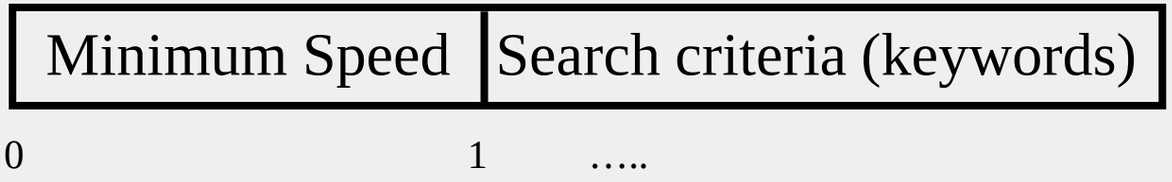
How do I search for my Beatles file?



Gnutella Message Header Format

How do I search for my Beatles file?

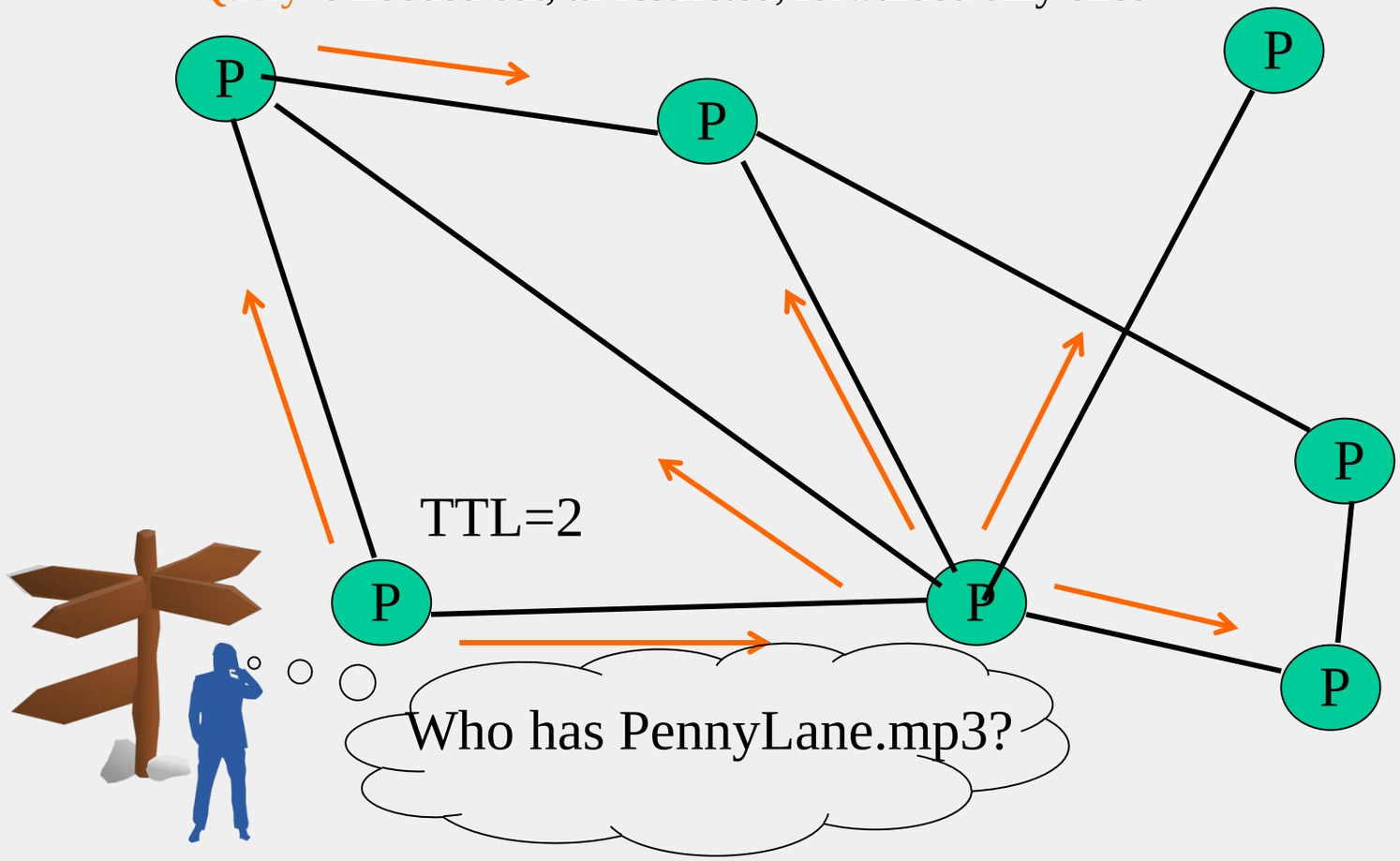
Query (0x80)



Payload Format in Gnutella Query Message

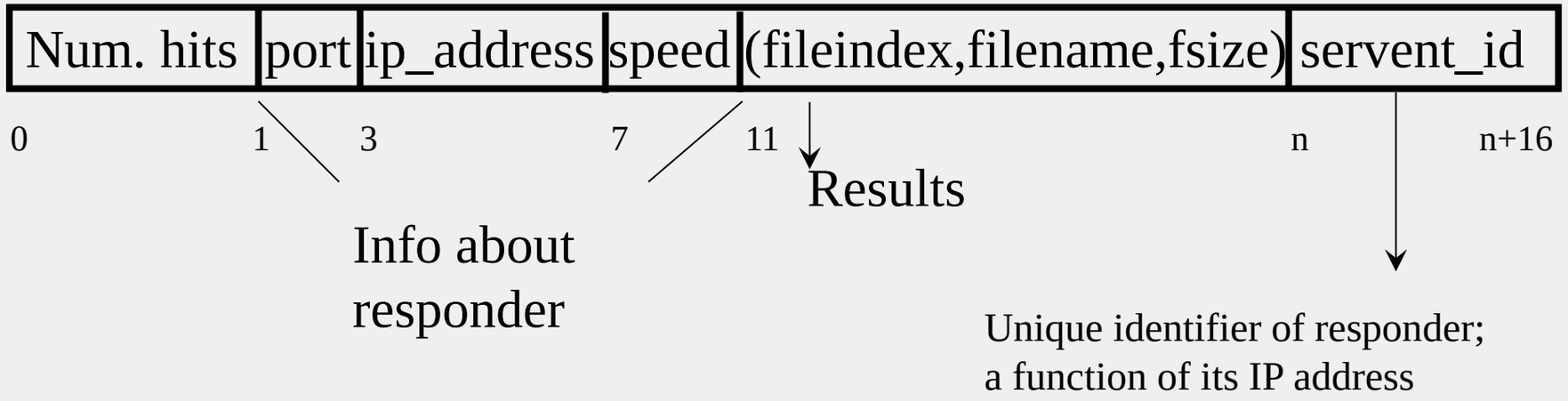
Gnutella Search

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



Gnutella Search

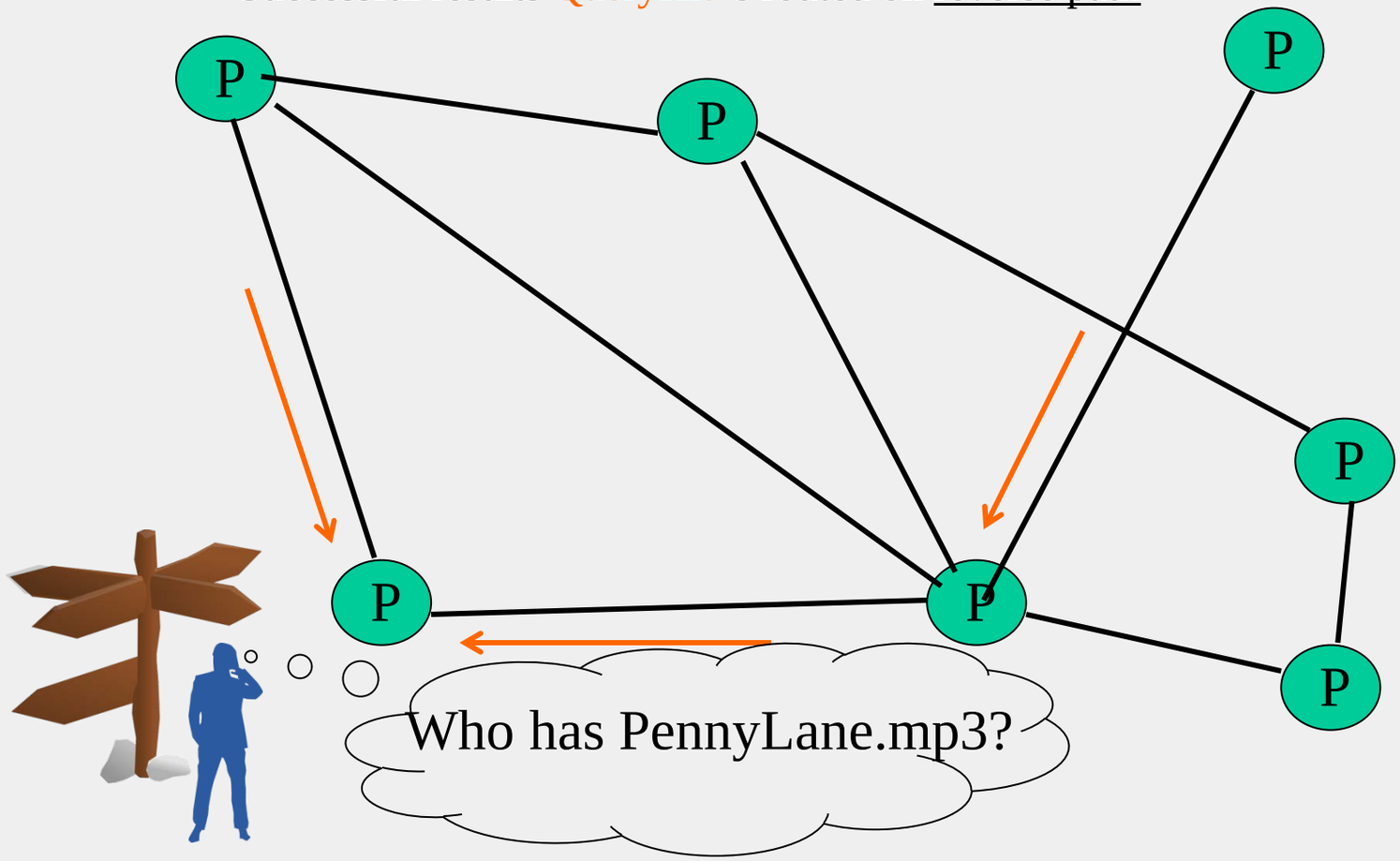
QueryHit (0x81) : successful result to a query



Payload Format in Gnutella QueryHit 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
- Duplicates with same DescriptorID and Payload descriptor (msg type) are dropped
- QueryHit with DescriptorID for which Query 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 with file packets after this message:

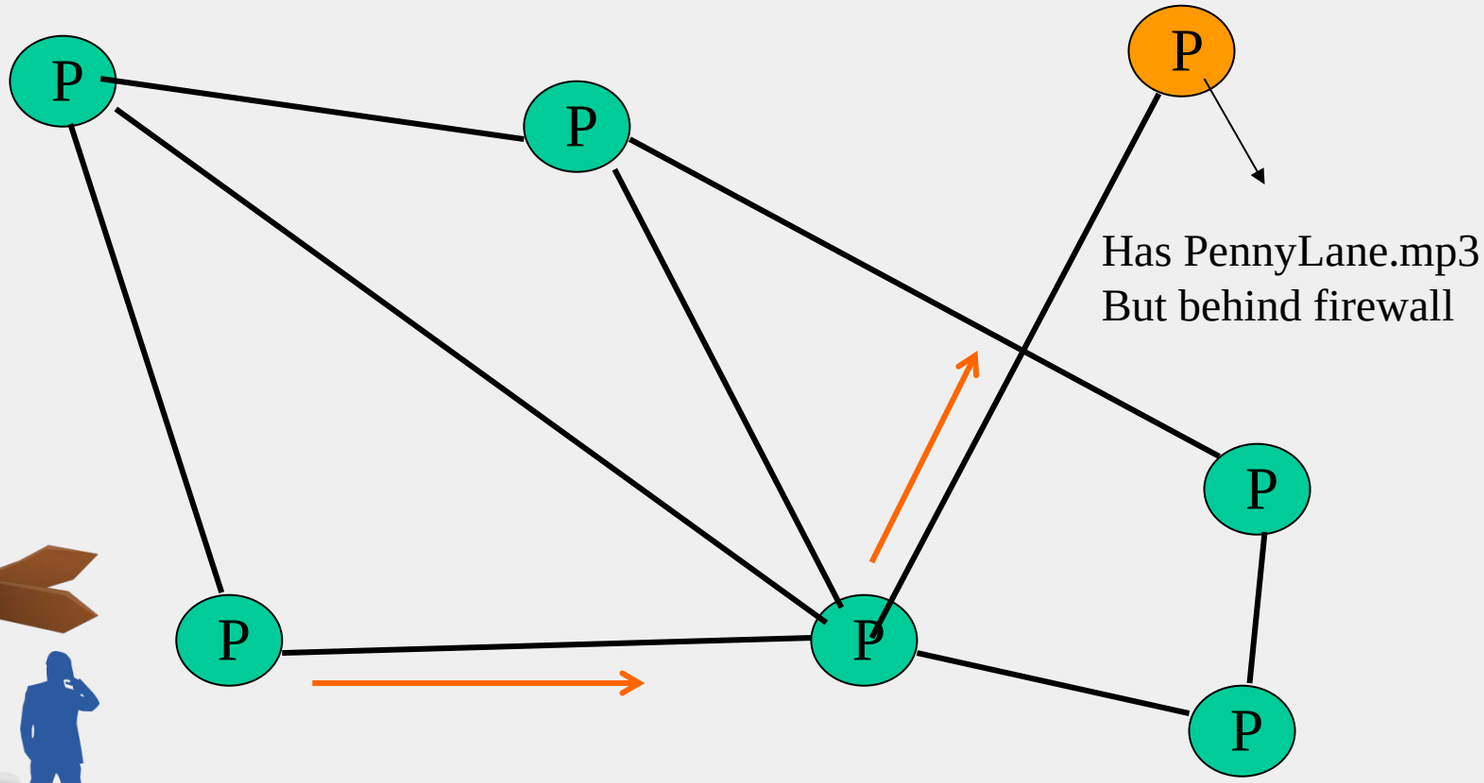
```
HTTP 200 OK\r\n
Server: Gnutella\r\n
Content-type:application/binary\r\n
Content-length: 1024 \r\n
\r\n
```

After receiving QueryHit messages (2)

- HTTP is the file transfer protocol. Why?
  - Because it's standard, well-debugged, and widely used.
- Why the "range" field in the GET request?
  - To support partial file transfers.
- What if responder is behind firewall that disallows incoming connections?

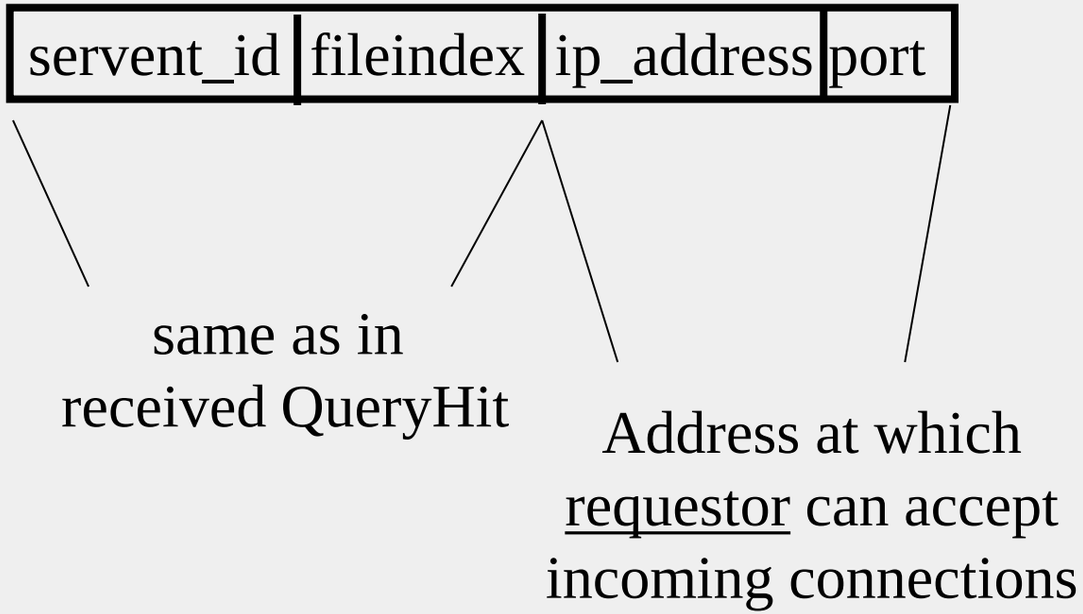
Dealing with Firewalls

Requestor sends Push to responder asking for file transfer



Dealing with Firewalls

Push (0x40)



Dealing with Firewalls

- 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 as explained earlier

- 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

- 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 peers joining, leaving and failing

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 (soon)

Problems (contd.)

- Large number of *freeloaders*
  - 70% of users in 2000 were freeloaders
  - Only download files, never upload own files
- 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 (coming up soon)