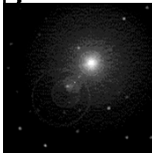


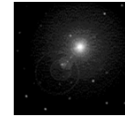
| | |
|---|---|
| | |
|  | <p>A Course on Foundations of Peer-to-Peer Systems & Applications</p> |
| | |

| | |
|---|--|
| | |
| <p>CS 6/75995 Foundation of Peer-to-Peer</p> <hr/> <p>Applications & Systems</p> | <p>Kent State University</p> <p>Dept. of Computer Science www.cs.kent.edu/~javed/class-P2P12/LECT-3</p> |
| | |

Emerging Classification Peer-to-Peer Systems

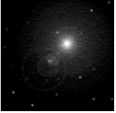
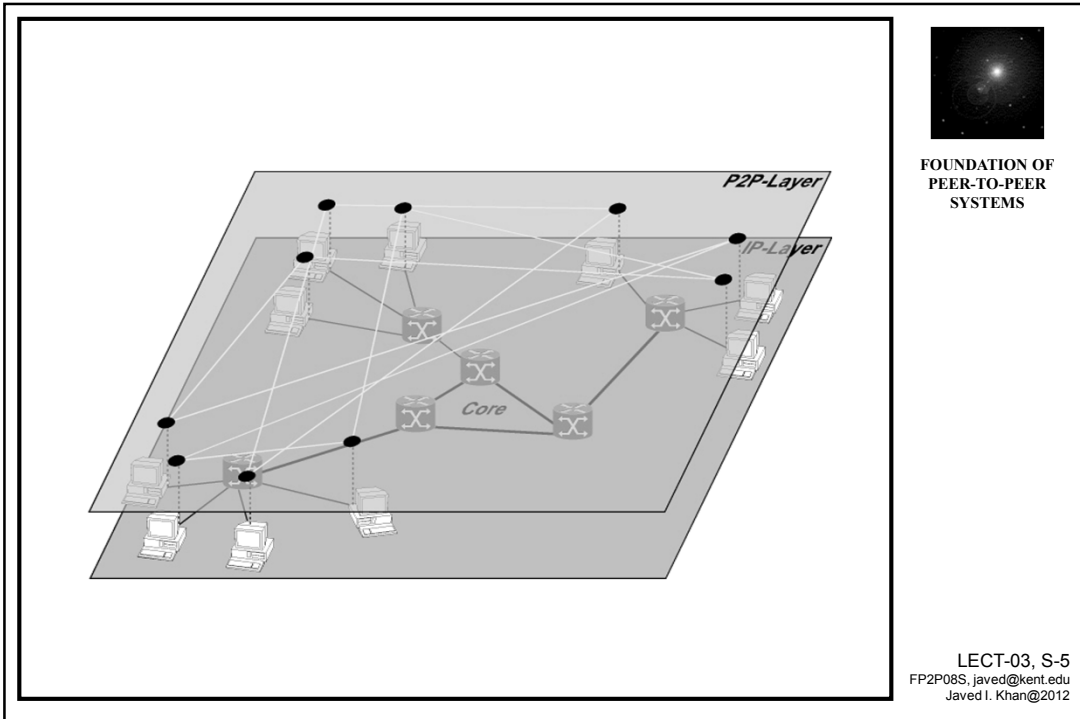
Some Basic Concepts

- Transport Network (Internet):
 - An internet transport network is made up of a set of internet hosts and physical links between the hosts, and acts as a transport network for a peer-to-peer system.
- Peer
 - A program element running on a subset of internet hosts.
- Overlay Network (P2P network)
 - A top level network defined on a transport network made up of peers and overlay links connecting the peers. An overlay link corresponds to a physical path (one or more physical links) of the underlying physical network.



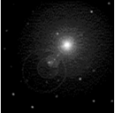
FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-4
FP2P08S, javed@kent.edu
Javed I. Khan@2012



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

- ### Lecture Overview
1. General Characteristics of Early Peer-to-Peer Systems
 2. Centralized Peer-to-Peer Networks
 1. Basic Characteristics
 2. Signaling Characteristics
 3. Discussion
 3. Pure Peer-to-Peer Networks
 1. Basic Characteristics
 2. Signaling Characteristics
 3. Discussion
 4. Hybrid Peer-to-Peer Networks
 1. Basic Characteristics
 2. Signaling Characteristics
 3. Discussion
- LECT-03, S-6
FP2P08S, javed@kent.edu
Javed I. Khan@2012

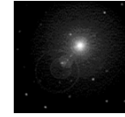


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

General Characteristics of 1st & 2nd Gen. P2P

1st and 2nd Generation P2P systems are overlay architectures, with the following characteristics:

- TCP/IP based
- Decentralized and self organizing (with possible centralized elements)
- Content:
 - Distributed “randomly” on the network, with several replicas
 - content and its descriptions are not structured (stays at the nodes which bring it into the network)
 - Content transfer:
 - Out of band, i.e. on separate connections and not via signaling connections
 - Mostly via HTTP
- Employ distributed shared resources (computing power and data storage)
- Generally two kinds of requests:
 - Content requests: to find content in the overlay
 - Keep-alive requests: stay connected in the overlay
- Initially developed for file-sharing
- Various realizations exist




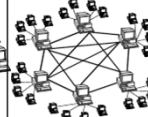



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-7
FP2P08S, javed@kent.edu
Javed I. Khan@2012

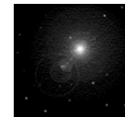
Classification

[*Eberspacher & Schollmeier 2005]

| Client-Server | Peer-to-Peer | | | |
|---|---|--|---|--|
| 1. Server is the central entity and only provider of service and content. → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system Example: WWW | 1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servent concept) | | | |
| | Unstructured P2P | | | Structured P2P |
| | <i>Centralized P2P</i> | <i>Pure P2P</i> | <i>Hybrid P2P</i> | <i>DHT-Based</i> |
| | 1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database Example: Napster | 1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities Examples: Gnutella 0.4, Freenet | 1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities Example: Gnutella 0.6, JXTA | 1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are “fixed” Examples: Chord, CAN |
|  |  |  |  |  |

1st Gen.

2nd Gen.

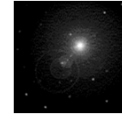


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-8
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Criticism of the Classification

- Centrality can be based on multiple service notion.
- Pure decentralization might not be a reality. It is more of attempt to minimize centrality and maximize decentrality of the of the core-operation needed to realize an architecture.

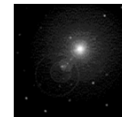


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-9
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Basic Topology Characteristics

- Peer:
 - Node actively participating in the overlay (PC running Kazaa)
 - content provider+content requestor+router in the overlay network
 - Identifiable by a General Unique ID (hash-value of “unique” or random ID)
- Overlay Topology: “virtual” signaling network established via TCP connections between the peers
- Characteristics of the overlay topology:
 - completely independent from physical network, because of the abstraction layer TCP/IP
 - May include hierarchies (rendezvous peers in JXTA) (hub network)
 - May include centralized elements (lookup server in Napster) (star network)
 - May be a completely randomized network (Gnutella 0.4) which can be modeled with random graphs (randomly meshed network)
 - Separate addressing scheme

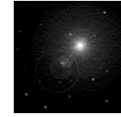


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-10
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Basic Routing Behavior

- Request messages:
 - Include a hop-counter, a GUID and a TTL (Time-to-Live) in the header
 - TTL determines along how many hops a message may be forwarded
 - Are flooded in the overlay network
 - Every node forwards every incoming message to all neighbors except the neighbor, it received the message from
 - Exceptions: see below
 - Request messages terminate, if
 - Same message-type with same GUID is received more than once (loop!!)
 - Hop-counter=TTL
- Response messages:
 - Include a hop-counter, a GUID and a TTL (Time-to-Live) in the header
 - GUID is the same as of the initializing request message
 - Are routed back on the same way to the requestor, the request message was transmitted to the responding peer
 - →every peer has to store the GUID of each request for a certain amount of time
 - No flooding to save resources

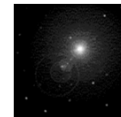


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-11
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Basic Bootstrapping

- Mostly not part of the protocol specification
- Necessary to know at least one active participant of the network
- Otherwise no participation at the overlay possible for a new node
- Address (TCP) of an active node can be retrieved by different means:
 - Bootstrap cache: Try to establish one after another a connection to a node seen in a previous session
 - Bootstrap server:
 - Connect to a "well known host", which almost always participates
 - Ask a bootstrap server to provide a valid address of at least one active node
 - Realizations:
 - FIFO of all node-addresses which recently used this bootstrap (a node which just connected is assumed to be still active)
 - Random pick of addresses which recently connected via this server to the overlay (+ no loops, -may be outdated)
 - Broadcast on the IP layer
 - Use multicast channels
 - Use IP broadcasting (-limited to local network)

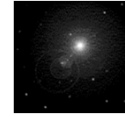


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

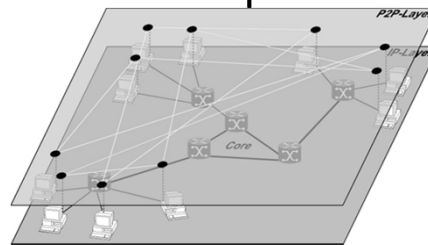
LECT-03, S-12
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Lecture Overview

1. General Characteristics of Early Peer-to-Peer Systems
2. Centralized Peer-to-Peer Networks
 1. Basic Characteristics
 2. Signaling Characteristics
 3. Discussion
3. Pure Peer-to-Peer Networks
 1. Basic Characteristics
 2. Signaling Characteristics
 3. Discussion
4. Hybrid Peer-to-Peer Networks
 1. Basic Characteristics
 2. Signaling Characteristics
 3. Discussion



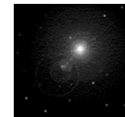
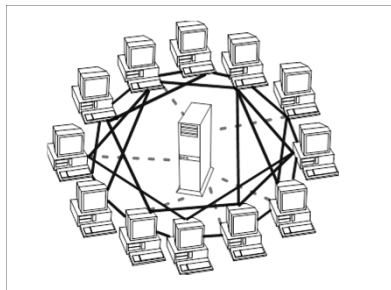
FOUNDATION OF
PEER-TO-PEER
SYSTEMS



FP2P08S, javed@kent.edu
Javed I. Khan@2012

Definition of centralized P2P

- All peers are connected to central entity
- Peers establish connections between each other on demand to exchange user data (e.g. mp3 compressed data)
- Central entity is necessary to provide the service
- Central entity is some kind of index/group database
- Central entity is lookup/routing table

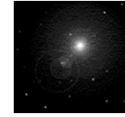


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-14
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Signaling Characteristics of Centralized Directory Based P2P

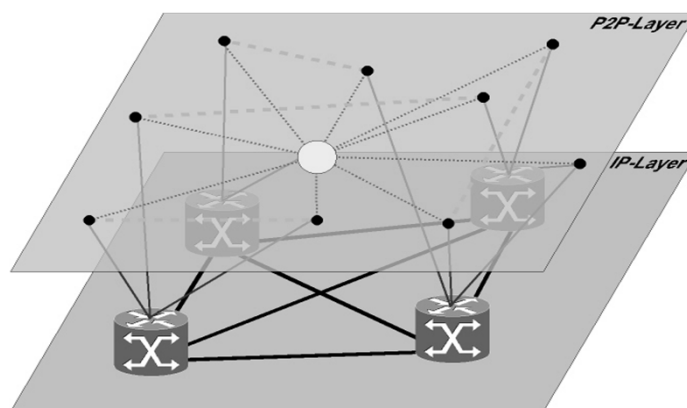
- Bootstrapping: Bootstrap-server = central server
- Central entity can be established as a server farm, but one single entry point = single point of failure (SPOF)
- All signaling connections are directed to central entity
- Peer ↔ central entity: P2P protocol, e.g. Napster protocol
 - To find content
 - To log on to the overlay
 - To register
 - To update the routing tables
 - To update shared content information
- Peer ↔ Peer: HTTP
 - To exchange content/data



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

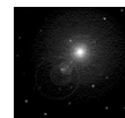
LECT-03, S-15
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Topology of Centralized P2P



Server
Connection between
2 servers (TCP)

Connection between
router & server
Connection between
routers (Core)



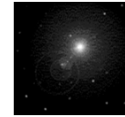
FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-16
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Example: Napster Architecture

Napster

- Program for sharing files over the Internet
- A “disruptive” application/technology?
- Brief History:
 - **May 1999**: Shawn Fanning (freshman, Northeastern University) founds Napster Online music service
 - **December 1999**: First Lawsuit
 - **March 2000**: University of Wisconsin reports that 25% of its IP traffic is Napster traffic
 - **December 2000**: estimated 60 million users
 - **February 2001**: US Circuit Court of appeals: Napster knew users violating copyright laws
→ Shut down of the service
 - The logo and brand name is now used by a music store.

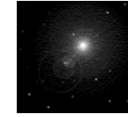
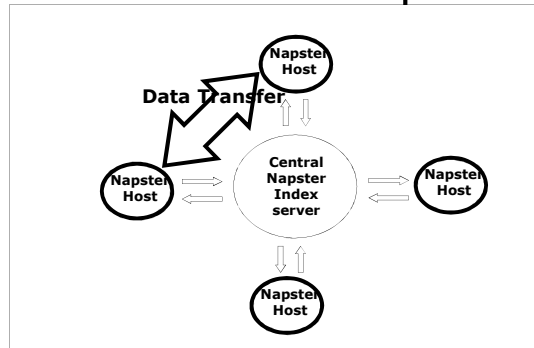


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-18
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Napster: How Does It Work

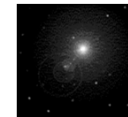
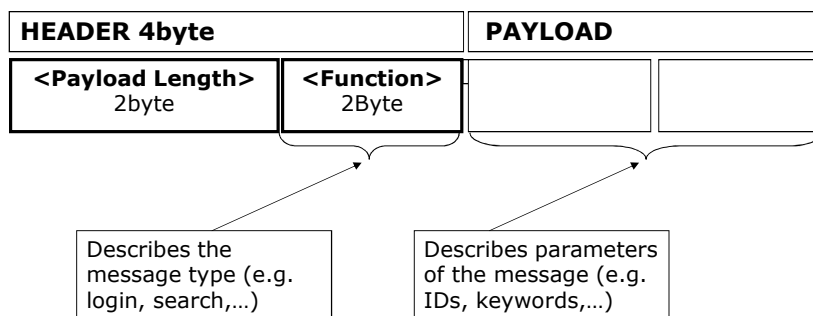
- Application-level, client-server protocol over point-to-point TCP
- Participants:
 - Napster Hosts/peers
 - Client Service
 - Login
 - Data-requests
 - Download-requests
 - P2P Service
 - Data-transfer
 - Napster Indexserver
 - Pure Server
- Five steps:
 - Connect to Napster Server
 - Upload your list of files (push) to server
 - Query Indexserver with a list of keywords to search the full list with
 - Select “best” of correct answers
 - Connect to providing host/peer



FOUNDATION OF
PEER-TO-PEER

LECT-03, S-19
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Napster Message Structure

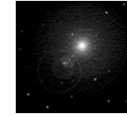
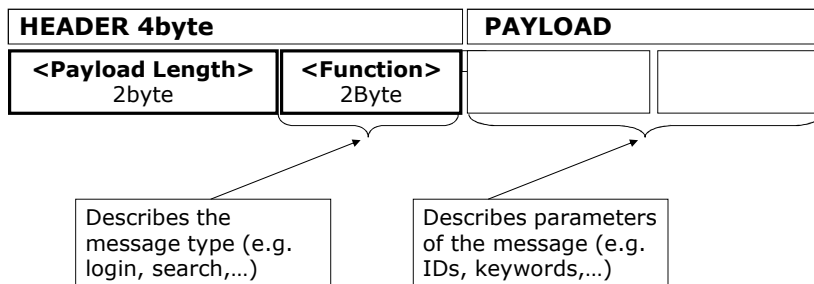


FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-20
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Message Format

- Napster uses TCP for client to server communication. Typically the servers run on ports 8888 and 7777. Note that this is different from the 'metaserver' (or redirector) which runs on port 8875.
- Each message to/from the server is in the form of
<length><type><data>



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-21
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Logging In

- **2 login [CLIENT]**

Format: <nick> <password> <port> "<client-info>" <link-type> [<num>]

<port> is the port the client is listening on for data transfer. If this value is 0, it means that the client is behind a firewall and can only push files outward. It is expected that requests for downloads be made using the 500 message (see below)
<client-info> is a string containing the client version info <link-type> is an integer indicating the client's bandwidth

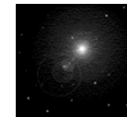
- 0 unknown
 - 1 14.4 kbps
 - 2 28.8 kbps
 - 3 33.6 kbps
 - 4 56.7 kbps
 - 5 64K ISDN
 - 6 128K ISDN
 - 7 Cable
 - 8 DSL
 - 9 T1
 - 10 T3 or greater
- <num> build number for the windows client [optional]

Example: foo badpass 6699 "nap v0.8" 3

- **3 login ack [SERVER]**

Format: <email>

the server sends this message to the client after a successful login (2). If the nick is registered, the <email> address given at registration time is returned. If the nick is not registered, a dummy value is returned.



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-22
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Initial Checks

- **4 version check [CLIENT]**

Format: <version>

Server sends [5] if an update is needed. All other responses are ignored.
<version> = string, version 2.0b5a sends "2.0"

- **5 "auto-upgrade" [SERVER]**

Format: <version> <hostname:filename>

Napster is out of date, get a new version. Or also known as gaping security hole.

<version> = string, the new version number.
<hostname> = string, hostname of upgrade (http) server
<filename> = string, filename

Connections are always made to port 80.

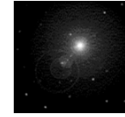
The HTTP Request:
GET <filename> HTTP/1.0
Connection: Keep-Alive
Host: <hostname>

Expected HTTP Response.
Content-Length: <size>
Content-Type: <doesn't matter>
<data>

Upgrade file is save as "upgrade.exe"
And executed as: upgrade.exe UPGRADE "<current.exe>"

As far as I can tell no confirmation is requested by Napster when it receives this message. And immediately start to "auto-upgrade". To keep users informed that Napster is doing something potentially very harmful to their computer it displays a message saying it's "auto-upgrading". I think this pretty bad, since all someone has to do is hack a napsterserver et voila a zillion clients at your disposal. As far as I know this only affects the Win32 2.0b5a Napster client. Other clients/versions I don't know.

[This section was contributed by Thomas van der Heijden <thom@bart.nl>-ed]



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-23
FP2P08S, javed@kent.edu
Javed I. Khan@2012

New User (1)

- **6 new user login [CLIENT]**

Format: <nick> <pass> <port> "<client-info>" <speed> <email-address>

This message is used when logging in for the first time to register a new nickname. The client normally checks for an unused nick using the "check nickname" (7) message and upon receipt of a "nickname not registered" (8) from the server will proceed to log in with this command.

note: this message is similar to the 0x02 message, with the addition of <email-address> on the end

Example: foo foo 6699 "nap v0.8" 3 email@here.com

- **7 nick check [CLIENT]**

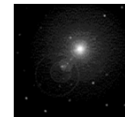
Format: <nick>

Queries the server to see if <nick> is already registered. This message is typically used prior to logging in for the first time to check for a valid nickname. Response to this message is one of 8, 9 or 10

- **8 nickname not registered [SERVER]**

Format: (empty)

The server sends this message in response to the "nick check" (7) message to indicate that the specified nickname is not already registered and is ok to use.



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-24
FP2P08S, javed@kent.edu
Javed I. Khan@2012

New User (2)

- **9 nickname already registered [SERVER]**

Format: (empty)

The server sends this message when the nickname the client has requested has already been registered by another user

- **10 (0x0a) invalid nickname [SERVER]**

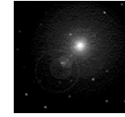
Format: (empty)

This server sends this message in response to the "check nickname" (7) messages when the specified nickname is invalid, usually due to bad characters such as spaces or other non-printable characters.

- **Accepted**

Format: (Character set)

- The Napster, Inc. servers only accept (upper- and lower-case) nicks comprised of the following: abcdefghijklmnopqrstuvwxyz1234567890_[]{}-@^!\$



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-25
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Sharing File

- **100 (0x64) client notification of shared file [CLIENT]**
Format: "<filename>" <md5> <size> <bitrate> <frequency> <time>

<md5> see section "MD5"
<size> is bytes
<bitrate> is kbps
<frequency> is hz
<time> is seconds

Example:

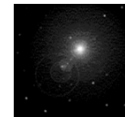
"generic band - generic song.mp3" b92870e0d41bc8e698cf2f0a1ddfac7 443332 128 44100 60

- **102 (0x66) remove file [CLIENT]**
Format: <filename>

client requests to remove file from shared library

- **110 unshare all files [CLIENT]**
Format: (empty)

Notifies the server that the client is no longer sharing any of the files previously shared.



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-26
FP2P08S, javed@kent.edu
Javed I. Khan@2012

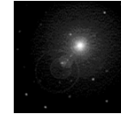
Sharing Directory

- 870 add files by directory [CLIENT]

Format: "<directory>" "<file>" <md5> <size> <bitrate> <freq>
<duration>

- This command allows a client to share multiple files in the same directory as a shortcut to multiple 100 (share file) commands.
- <directory> is the path which should be prepended to all the given filenames in the rest of the command.
- <file> is the name of the file to share *without* its directory component. When other clients do a browse/share, the real path will be reported as <directory>/<file>.

The portion of this command after the <directory> may be repeated as many times as necessary for files in the same directory. NOTE: most servers will not accept commands of length longer than 2048 bytes so you still may need to break up the files into multiple commands if there are many files in a single directory.



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-27
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Query (1)

- 200 (0xc8) client search request [CLIENT]

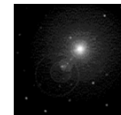
[FILENAME CONTAINS "artist name"] MAX_RESULTS <max>
[FILENAME CONTAINS "song"] [LINESPEED <compare> <link-type>]
[BITRATE <compare> "
"] [FREQ <compare> "<freq>"] [WMA-
FILE] [LOCAL_ONLY]

- The artist name and the song name are, obviously, treated the same by the server; confirm this for yourself on the windows client. max is a number; if it is greater than 100, the server will only return 100 results.

<compare> is one of the following: "AT LEAST" "AT BEST" "EQUAL TO"
<link-type> see 0x02 (client login) for a description

 is a number, in kbps
<freq> is a sample frequency, in Hz
LOCAL_ONLY causes the server to only search for files from users on the same server rather than all linked servers.
- The windows client filters by ping time inside the client. It pretty much has to, and it's easy to see the result by setting ping time to at best 100 ms or so, and max search terms to 50. You'll get back like 3 results, but the client will still tell you that it found "50 results".

Examples:
FILENAME CONTAINS "Sneaker Pimps" MAX_RESULTS 75 FILENAME CONTAINS "tesko suicide"
BITRATE "AT LEAST" "128" MAX_RESULTS 100 FILENAME CONTAINS "Ventolin" LINESPEED
"EQUAL TO" 10



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

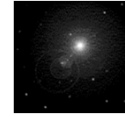
LECT-03, S-28
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Query Response (2)

- 201 (0xc9) search response [SERVER]**
**"<filename>" <md5> <size> <bitrate> <frequency> <length>
 <nick> <ip> <link-type> [weight]**
 - <md5> see section "MD5"
 <size> is file size in bytes
 <bitrate> is mp3 bit rate in kbps
 <frequency> is sample rate in hz
 <length> is the play length of the mp3 in seconds
 <nick> the person sharing the file
 <ip> is an unsigned long integer representing the ip address of the user with this file <link-type> see message client_login (2) message for a description [weight] a weighting factor to allow the client to sort the list of results. positive values indicate a "better" match, negative values indicate a "worse" match. If not present, the weight should be considered to be 0.

Example:

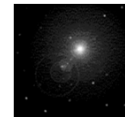
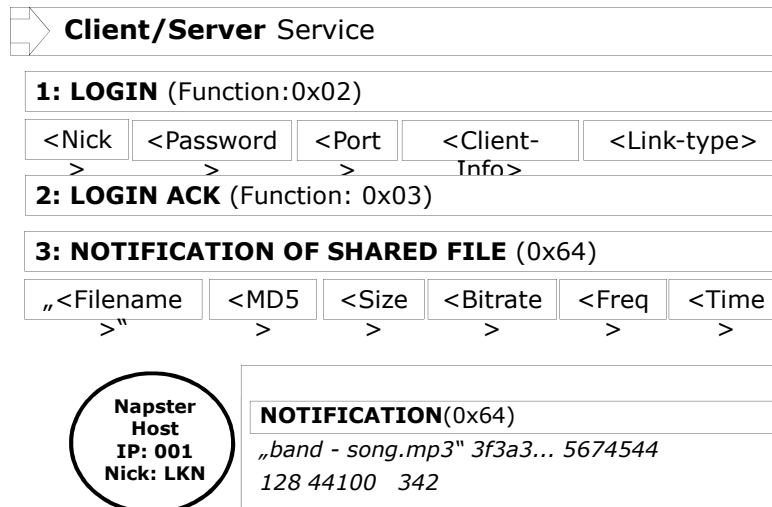
```
"random band - random song.mp3" 7d733c1e7419674744768db71bf88bed 2558199 128 44100 159 lefty
3437166285 4
```
- 202 (0xca) end of search response from server [SERVER]**
Format: (empty)



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-29
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Napster: Initialization



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-30
FP2P08S, javed@kent.edu
Javed I. Khan@2012

Napster: File Request Procedure

1: SEARCH (Function: 0xC8)

[FILENAME CONTAINS „Search Criteria“] [MAX_RESULT <Max>]

[LINESPEED <Compare> <Link-Type>]

[BITRATE <Compare> „<Bitrate>“] [FREQ <Compare> „<Freq>“]

2: SEARCH RESPONSE (Function: 0xC9)

„<Filename>“ <MD5> <Size> <Bitrate> <Freq>

<Time> <Nick> <IP> <Link-Type>



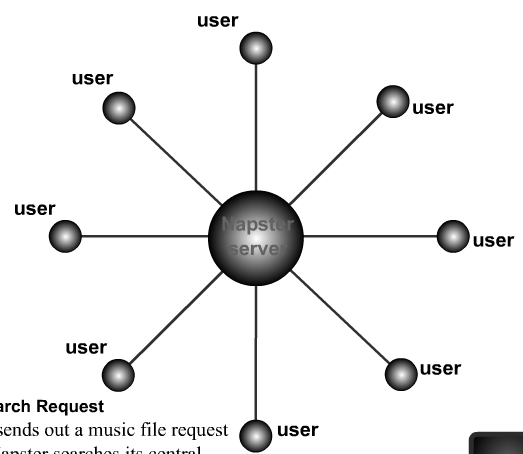
**Central
Napster
Index
server**

SEARCH(0xC8)
 FILENAME CONTAINS „song“ MAX_RESULTS 100
 LINESPEED „AT LEAST“ 6 BITRATE „AT LEAST“ „128“
 FREQ „EQUAL TO“ „44100“

**Napster
Host
IP: 002
Nick: MIT**

LECT-03, S-31
 FP2P08S, javed@kent.edu
 Javed I. Khan@2012

Napster: Basics

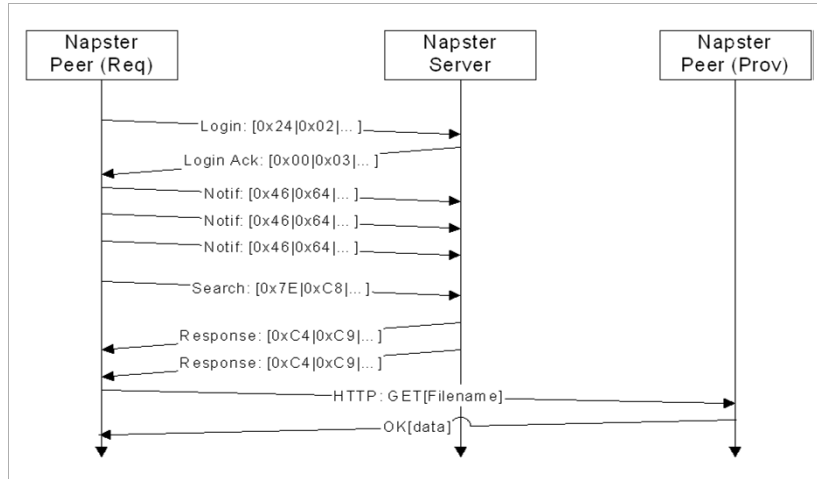


1) Search Request
 User sends out a music file request
 and Napster searches its central
 data base

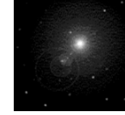


LECT-03, S-32
 FP2P08S, javed@kent.edu
 Javed I. Khan@2012

Napster Messages



Sample message sequence chart for one Napster server with one requesting and one providing peer



FOUNDATION OF
PEER-TO-PEER
SYSTEMS

LECT-03, S-33
FP2P08S, javed@kent.edu
Javed I. Khan@2012