Peer to Peer Networking Seminar Series

Napster

By

Iqbal Mohomed & Carl Nehme

Topics

- I: An Introduction to Napster
- II: Architecture/ Protocol
- III: Performance Bottlenecks
- IV: Comparision

I: An Introduction to Napster

An overview of the
Service/Application

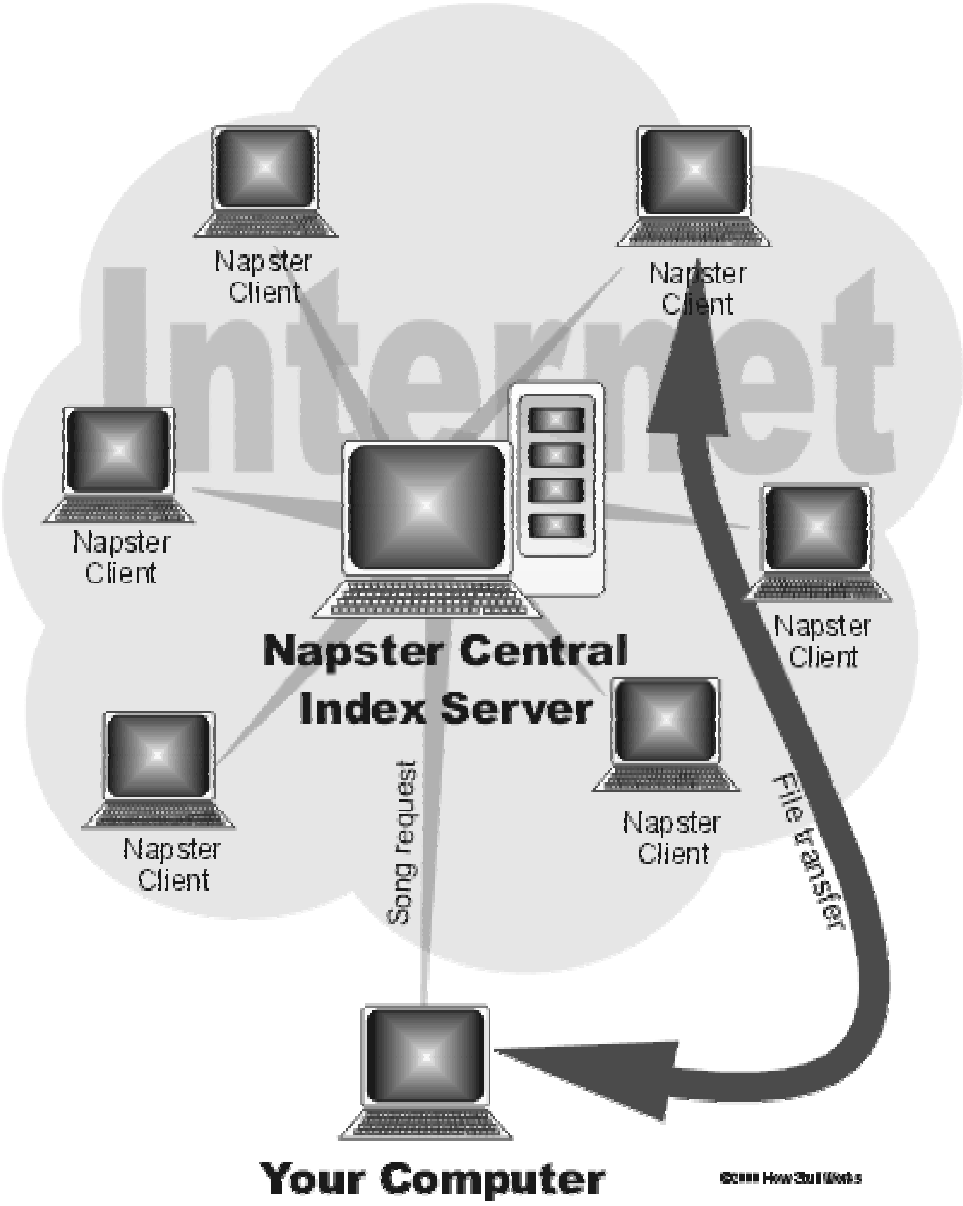
- The advantage of the MP3 format is that it makes song files small enough to move around on the Internet in a reasonable amount of time.

- The initial MP3 craze was fueled by sites like mp3.com. On these sites, anyone can upload a song. The songs are then stored on a server that is part of the Web site. Other users can connect to the Web site and download songs they are interested in. Another way of obtaining MP3 files is to perform a search on the title or artist that you are looking for. Quite often, the search would return a lot of links that were **broken**, meaning that the Web page could not be found.

In early 1999, Shawn Fanning began to develop an idea as he talked with friends about the difficulties of finding the kind of MP3 files they were interested in. He thought that there should be a way to create a program that combined three key functions into one. These functions are:

- **Search engine:** Dedicated to finding MP3 files only
- **File sharing:** The ability to trade MP3 files directly, without having to use a centralized server for storage
- **Internet Relay Chat (IRC):** A way to find and chat with other MP3 users while online Fanning, only 18 at the time, spent several months writing the code that would become the utility **Napster**. He uploaded the original beta version to download.com, where it quickly became one of the hottest downloads on the site.

Napster (Napster was Fanning's nickname in high school, because of his hair) is a different way to distribute MP3 files. Instead of storing the songs on a central computer, the songs live on users' machines. This is called **peer-to-peer sharing**, or P2P. When you want to download a song using Napster, you are downloading it from another person's machine, and that person could be your next-door neighbor or someone halfway around the world.



Let's take a look at what is necessary for you to download a song that you are interested in:

You need:

•A copy of the Napster utility installed on your computer.

•A directory on your computer that has been **shared** so that remote users can access it.

•Some type of Internet connection.

The provider of the song needs:

•A copy of the Napster utility installed on their computer.

•A directory on their computer that has been **shared** so that you can access it.

•Some type of Internet connection that is currently on.

•A copy of the song you are looking for in the designated shared directory.

Here is what happens when you decide to look for the song:

1. You open the Napster utility.
2. Napster checks for an Internet connection.
3. If it finds a connection, Napster logs you onto the central server. The main purpose of this central server is to keep an index of all the Napster users currently online and connect them to each other. It does not contain any of the MP3 files itself.
4. You type in the title or artist of the song you are looking for.
5. The Napster utility on your computer **queries** the index server for other Napster computers online that have the song you have requested.
6. Whenever a match is found, the Napster server informs your computer where to find the requested file.
7. When the server replies, Napster builds a list of these systems in the results window.
8. You click on the file(s) that interest you and then choose **Download**.
9. Your copy of Napster attempts to establish a connection with the system hosting the file you have selected.
10. If a connection is successfully made, the file begins downloading.
11. Once the file is downloaded, the host computer breaks the connection with your system.
12. You open up your MP3 player software and listen to the song.

Where this works:

Simple implementations that usually work for regular people. Complex instructions will cut usage. For example, setting up Napster is easy, while Gnutella (which doesn't use a centralized organizer) may need lots of understanding of IP addresses, routing, and firewalls. The fact that Napster has a special client-based program for uploading, like many photo web sites, is what makes it easy, not that it's P2P.

The **same data on many different PCs**. If only one PC has the data, access to it could be unreliable.

The **files are static**, the information being downloaded is never changed. The files shared with Napster are not news feeds -- they are more likely the works of dead musicians.

Data such that you don't mind trusting the person sharing it.

Lots of college students with desktops connected to local ethernets.

Where it probably doesn't work as well:

Unique content on each PC where reliability or constant availability is important. When I want your pictures I don't want to have to call you up and have you boot up your PC. (Like when you have one phone line and an old fax: call first, then tell me to connect the fax -- if I'm home -- and then call back...). Many of us live with laptops which are only connected during the day and infrequently (hopefully) at night.

Content that constantly changes. This makes the copies people recently downloaded obsolete, and effectively gets back to only one copy on one specific PC, except you may not know it.

Content that requires a trust relationship with the source. Because of the "search everywhere" nature and simple signup of Napster-style P2P, *caveat emptor*.

Reliable connection speed. The data you want may be on a low-speed link. Your T1 doesn't help much.

II: Architecture/ Protocol

Client Server with a twist

- Napster is a TCP/IP Client-Server application where the clients also act as servers (Terminology note: we continue to call these clients).
- The Napster server acts as a directory or broker. When you download a file, this is done through a client-client connection.

Connection-oriented Protocol

- Napster uses a proprietary connection-oriented protocol.
- A “Napster session” begins with username/password authentication. Once completed, this does not have to be repeated.
- “Napster Sessions” are fairly long. Average length: 30 minutes for dialup connections, 5 hours hours for T1 and higher connections.
- “Download sessions” exist between two clients. These last no longer than the duration of the file download.

Downloading Protocol

- There are two types of downloads.
- “Normal downloads” are when the source client connects to the destination client.
- “Firewall downloads” are when the destination client connects to the source.
- In both cases, Napster is only the broker. It tells the destination to set up a transmission port. This information is then relayed to the source client.

Centralized Servers

- This is the part where Napster deviates from pure peer-to-peer. Ironically, this architecture made them successful and also brought about their downfall.
- Napster uses centralized servers for three logical functions: Redirection, Authentication and Directory Services.
- During a Napster session, the client only sees two Napster servers. The Authentication and Directory Services server are the same.

Redirection Server

- AKA Metaserver. This runs on port 8875. This is the first server the client connects to. This basically gives the client the IP address and port of a different server. Useful for load-balancing, maintenance, redundancy, etc.
- Napster has multiple Redirection Servers (atleast 3).

Authentication Server

- As noted, there are multiple instances of these on different ports (typically 8888 and 7777).
- Client sends:

<nickname> <password> <port> "<client-info>" <link-type> [<num>]

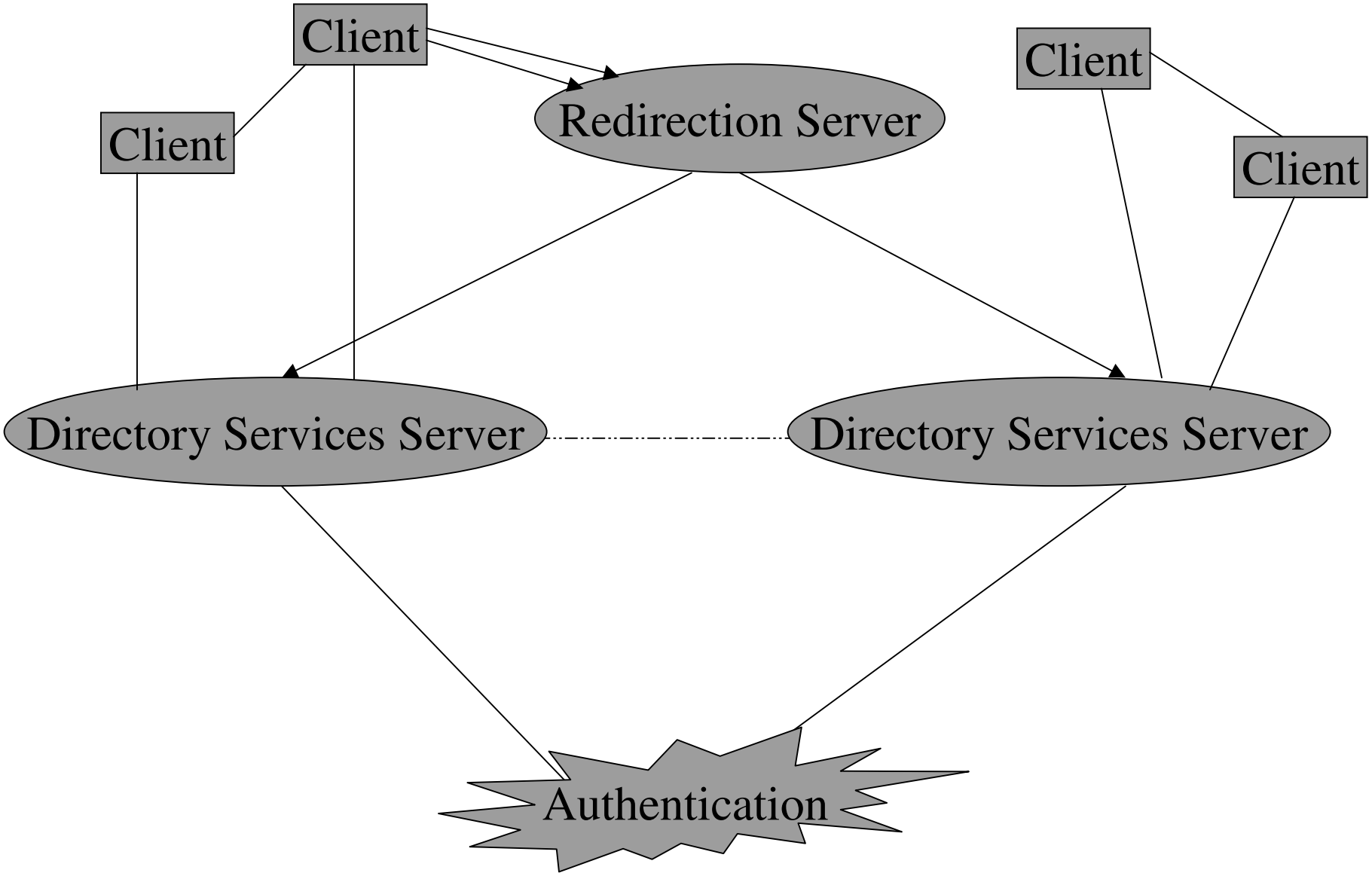
- Server responds:

<email>

If nickname is registered, the email at registration is returned. Otherwise a dummy value is used.

Directory Services Server

- When you search for a file, this is done on Napster's Directory Services server. Other things like Hotlist and Chat are done through this as well.
- There are a number of Directory Services servers that are clustered together. This is NOT transparent. Many operations depend on which DS you connect to. Infact, these are controlled parameters that Napster sets. For example, does your search occur only on your DS or on all.


Client
Client
Redirection Server
Client
Client
Directory Services Server
Directory Services Server
Authentication

Redundancy

- Keeping DS's separated as islands is good because if something goes wrong on one, a user can reconnect to another. The redirector is critical in this aspect as the user cannot control which DS they connect to from the Napster client.
- When you download a file, it becomes available for other people to download from you. Thus, the system has a built in redundancy.

Namespaces

- Napster establishes it's own namespace. Users must register a Nickname before they are allowed to use the service.
- The Nicknames are allocated on a first-come-first-serve basis.
- During a Napster session, the DS keeps track of the association between the Nickname and the client's IP address.
- At registration time, the user must provide a valid email address and possibly some demographic information.
- Namespace must be shared across all DS's.

III: Performance Bottlenecks

Issues

- Napster is the ultimate bandwidth hog. Many universities network traffic increased by 60-80%.
- As a system, there is a lot of redundancy because many people store the same file. This can be viewed as a performance issue with the system. Note that this is an uncontrollable parameter.
- Activity on DSs may become very high. This causes admins to isolate them. This reduces the efficiency of the system.

IV: Comparision

Napster VS Gnutella

Napster

- Uses centralized Directory Servers. + faster, efficient, easy to use
- Proprietary protocol
- Single client

Gnutella

- More decentralized. + Can't be shut down
- "Open" standard. + Allows people to develop clients/ services on top.
- Multiple front-ends available

References

- NapDev discussion goup on Yahoo
- Protocol specification at http://mynapster.sourceforge.net/webclient/napster.txt
- David Weekly's web page (disappeared?)
- Text: Peer to Peer Networking
- Marshall Brain's "How Stuff Works"/ http://www.howstuffworks.com