Napster and Freenet
Peer-to-Peer File Storage

Emin Gun Sirer

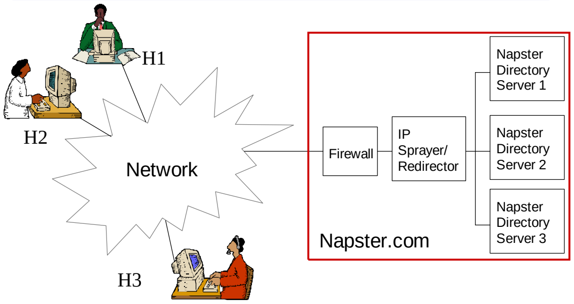
Napster

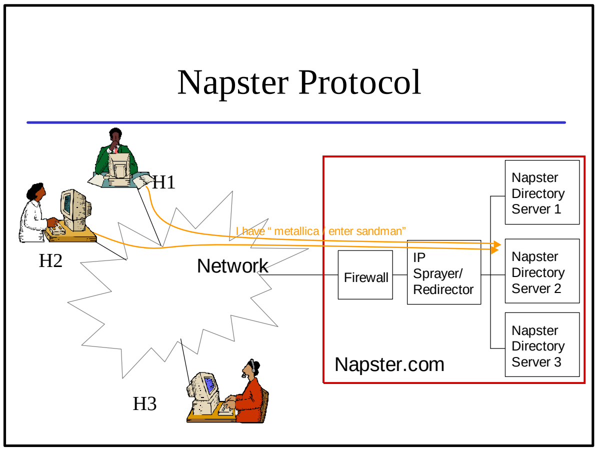
- Flat filesystem
  - Single-level filesystem with no hierarchy
  - Can have multiple files with the same name
- All storage is done at the edges
  - Each host computer exports a set of files that reside locally on that host.
  - The host is registered with a centralized directory; uses keepalives to show that it is still connected
  - A centralized directory is notified of the filenames that are exported by that host
- Simple, centralized directory

Napster Directory

- File lookup in Napster
  - Client queries directory server for filenames matching a pattern
  - Directory server picks 100 files that match the pattern, sends them to the client
  - Client pings each, computes round trip time to each host, displays results
  - User then transfers file directly from the closest host
- File transfers are peer-to-peer, with no involvement of anyone other than the two edge hosts

Napster Architecture


H1
H2
H3
Network
Firewall
IP Sprayer/ Redirector
Napster Directory Server 1
Napster Directory Server 2
Napster Directory Server 3
Napster.com



Napster Protocol

H1

H2

H3

I have " metallica / enter sandman"

Network

Firewall

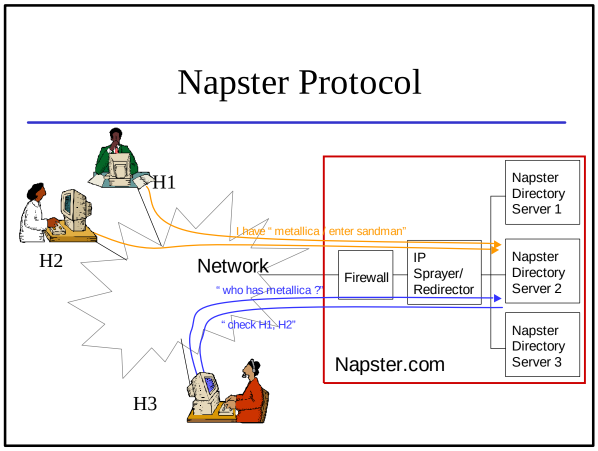
IP Sprayer/ Redirector

Napster Directory Server 1

Napster Directory Server 2

Napster Directory Server 3

Napster.com



Napster Protocol

H1

H2

H3

I have " metallica / enter sandman"

Network

" who has metallica ?"

" check H1, H2"

Firewall

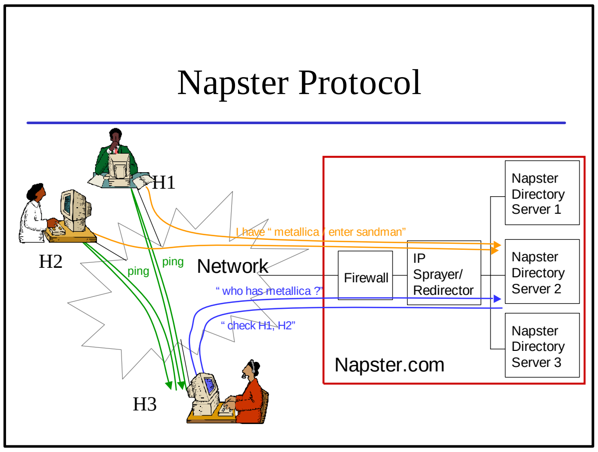
IP Sprayer/ Redirector

Napster Directory Server 1

Napster Directory Server 2

Napster Directory Server 3

Napster.com



Napster Protocol

H1

H2

H3

I have " metallica / enter sandman"

ping

ping

Network

" who has metallica ?"

" check H1, H2"

Firewall

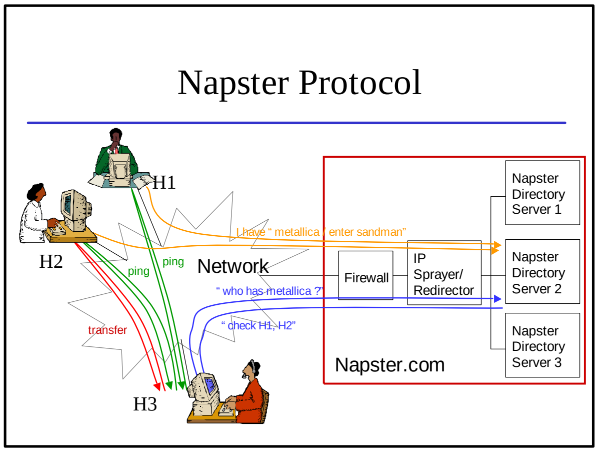
IP Sprayer/ Redirector

Napster Directory Server 1

Napster Directory Server 2

Napster Directory Server 3

Napster.com



Napster Protocol

H1

H2

H3

I have " metallica / enter sandman"

ping

ping

transfer

Network

" who has metallica ?"

" check H1, H2"

Firewall

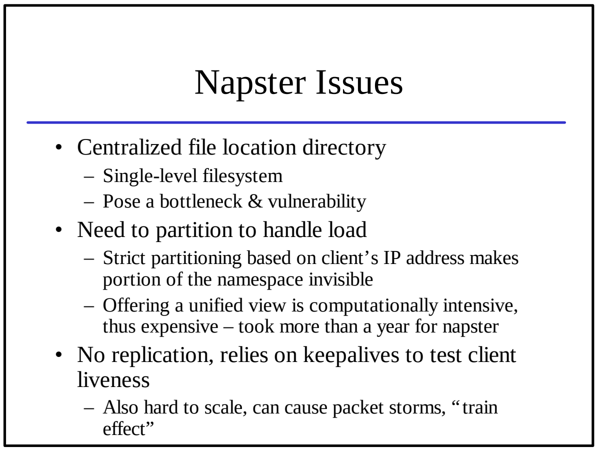
IP Sprayer/ Redirector

Napster Directory Server 1

Napster Directory Server 2

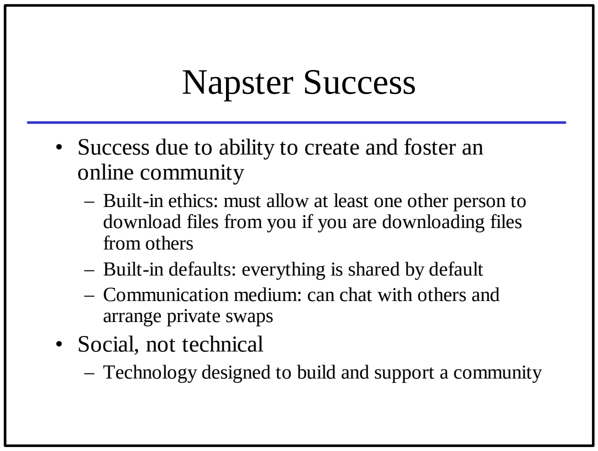
Napster Directory Server 3

Napster.com



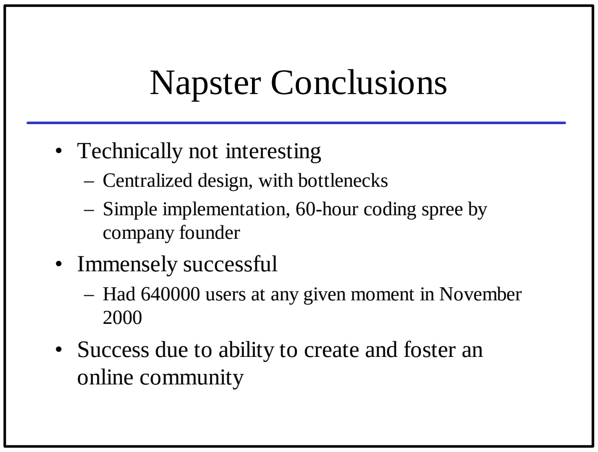
Napster Issues

- Centralized file location directory
  - Single-level filesystem
  - Pose a bottleneck & vulnerability
- Need to partition to handle load
  - Strict partitioning based on client’s IP address makes portion of the namespace invisible
  - Offering a unified view is computationally intensive, thus expensive – took more than a year for napster
- No replication, relies on keepalives to test client liveness
  - Also hard to scale, can cause packet storms, “train effect”



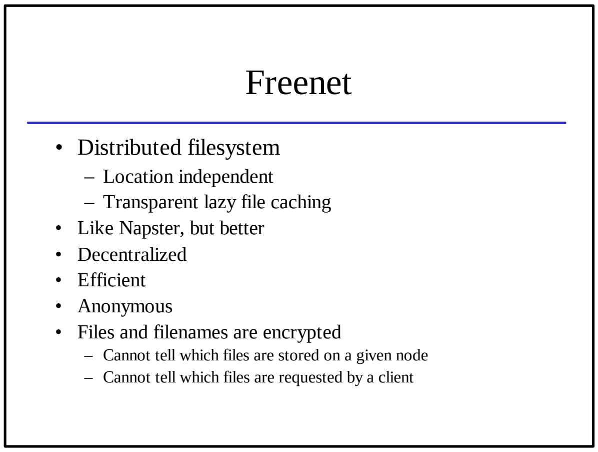
Napster Success

- Success due to ability to create and foster an online community
  - Built-in ethics: must allow at least one other person to download files from you if you are downloading files from others
  - Built-in defaults: everything is shared by default
  - Communication medium: can chat with others and arrange private swaps
- Social, not technical
  - Technology designed to build and support a community



Napster Conclusions

- Technically not interesting
  - Centralized design, with bottlenecks
  - Simple implementation, 60-hour coding spree by company founder
- Immensely successful
  - Had 640000 users at any given moment in November 2000
- Success due to ability to create and foster an online community



Freenet

- Distributed filesystem
  - Location independent
  - Transparent lazy file caching
- Like Napster, but better
- Decentralized
- Efficient
- Anonymous
- Files and filenames are encrypted
  - Cannot tell which files are stored on a given node
  - Cannot tell which files are requested by a client

Freenet Design Goals

- Anonymity for both producers and consumers of information
- Deniability for storers of information
  - Node operators are protected from legal entanglements because they cannot know what is stored on their machine
- Resistance to attempts by third parties to deny access to information
  - Malicious users cannot make other people’s files inaccessible
- Efficient dynamic storage and routing of information
  - Hash-based routing
- Decentralization of all network functions
  - No central bottlenecks

Freenet Structure

- Graph topology
- Storage nodes go online, attach themselves to other arbitrary nodes
- Users can treat the whole collection as a single, monolithic, global storage system

Freenet Naming

- Hierarchical namesystem
  - Files are identified by the hash of their filenames
  - Cannot have multiple files with the same name
  - Global single-level namespace is not desirable, since malicious users can engage in “key-squatting”
- Two-level namespace
  - Each user has their own directory

Freenet File Export

- Consider exporting file with name “Sun Tzu, Art of War”
  - Compute a public/private key pair from name using a deterministic algorithm
- File is encrypted with the hash of the public key
  - Goal is not to protect data – the file contents should be visible to anyone who knows the original keyword
  - Goal is to protect site operators – if a file is stored on your system, you have no way of decrypting its contents
- File is signed with the private key
  - Integrity check (though not a very strong one)

Freenet Namespace

- To allow others to read the file, need only publicize its original name
  - If you know “Sun Tzu, Art of War,” can trivially compute the key used to encrypt the file
  - Otherwise, decrypting requires reversing a one-way hash function
- This structure forms a flat global namespace
  - Nothing prohibits separate users from choosing the same name for different files
  - Or they could squat on keys – pick a common descriptive name, e.g. “Metallica,” export a sermon on washed-out rock stars

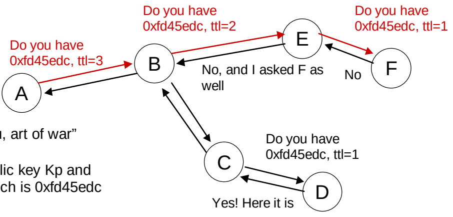
Freenet Directories

- Two-level directories
  - Users can create a signed-subspace
  - Akin to creating a top-level directory per user
  - Subspace creation involves generating a public/private key pair for the user
  - The user’s public key is hashed, XORed and then rehashed with the file public key to yield the file encryption key
- For retrieval, you need to know the user’s public key and the file’s original name

Locating Files in Freenet

- File producers publicize the original names of their files in a public forum
  - Web pages, forums, web spiders search engines
- Consumers acquire or compute the file key
  - Need original name and subspace public key
- Consumer asks nearest Freenet node for a copy of the file with given key
  - The request has a “hops-to-live” field
- Depth-first search is used to locate the file
  - File is cached locally if found

Freenet Query

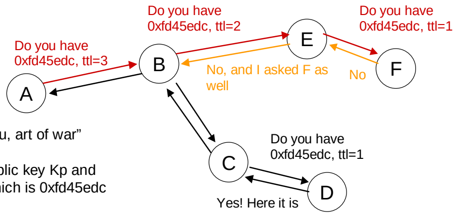


Want “sun tzu, art of war”

Compute public key Kp and Hash(Kp) which is 0xfd45edc

Asks B for file with hash 0xfd45edc, ttl=4
B asks E, then C
C retrieves it from D
A caches file

Freenet Query



Do you have 0xfd45edc, ttl=3

Do you have 0xfd45edc, ttl=2

Do you have 0xfd45edc, ttl=1

No, and I asked F as well

No

Do you have 0xfd45edc, ttl=1

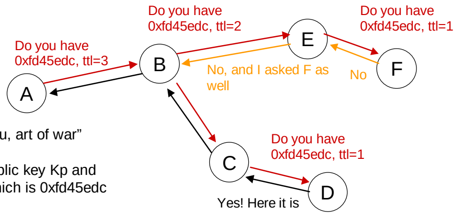
Yes! Here it is

Want “sun tzu, art of war”

Compute public key Kp and Hash(Kp) which is 0xfd45edc

Asks B for file with hash
0xfd45edc, ttl=4
B asks E, then C
C retrieves it from D
A caches file

Freenet Query



Do you have 0xfd45edc, ttl=3

Do you have 0xfd45edc, ttl=2

Do you have 0xfd45edc, ttl=1

No, and I asked F as well

No

Do you have 0xfd45edc, ttl=1

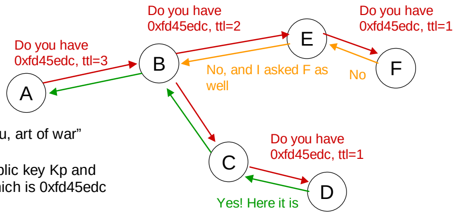
Yes! Here it is

Want “sun tzu, art of war”

Compute public key Kp and Hash(Kp) which is 0xfd45edc

Asks B for file with hash
0xfd45edc, ttl=4
B asks E, then C
C retrieves it from D
A caches file

Freenet Query



Do you have 0xfd45edc, ttl=3

Do you have 0xfd45edc, ttl=2

Do you have 0xfd45edc, ttl=1

No, and I asked F as well

No

Do you have 0xfd45edc, ttl=1

Yes! Here it is

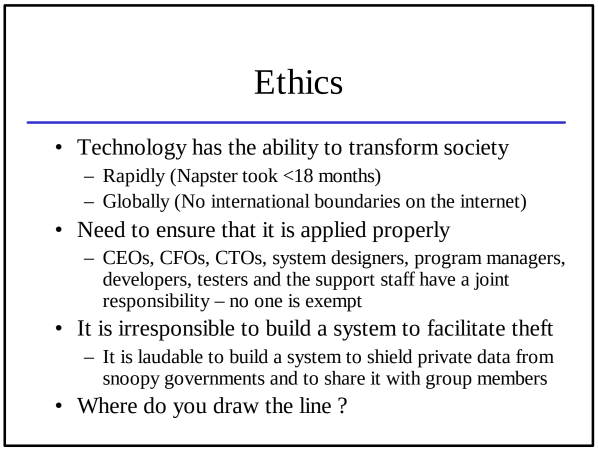
Want “sun tzu, art of war”

Compute public key Kp and Hash(Kp) which is 0xfd45edc

Asks B for file with hash
0xfd45edc, ttl=4
B asks E, then C
C retrieves it from D
A caches file

Loose Freenet Semantics

- Queries for similar keys are routed to the same hosts
  - Hence those hosts become experts at serving a portion of the namespace
- Any message can be dropped after a threshhold period
  - Client required to requery the system if it doesn’ t get an answer
- Any host along the path may alter the originator name in the query and place its own name
- Data copied from source to sink with every successful query and cached



Ethics

- Technology has the ability to transform society
  - Rapidly (Napster took <18 months)
  - Globally (No international boundaries on the internet)
- Need to ensure that it is applied properly
  - CEOs, CFOs, CTOs, system designers, program managers, developers, testers and the support staff have a joint responsibility – no one is exempt
- It is irresponsible to build a system to facilitate theft
  - It is laudable to build a system to shield private data from snoopy governments and to share it with group members
- Where do you draw the line ?